Títol: **GESCONDA II: INTEGRACIÓ DE COMPONENTS, REDISSENY, REIMPLEMENTACIÓ I AMPLIACIÓ DE FUNCIONALITAT**

Volum:	**1 / 1**
Alumne:	**OLM MARGARIT, ESTEVE**
Director/Ponent:	**SÀNCHEZ i MARRÈ, MIQUEL**
Departament:	**LSI**
Data:	**4 DE JULIOL DE 2007**

DADES DEL PROJECTE

Títol del Projecte:	**GESCONDA II: INTEGRACIÓ DE COMPONENTS, REDISSENY, REIMPLEMENTACIÓ I AMPLIACIÓ DE FUNCIONALITAT**
Nom de l'estudiant:	**OLM MARGARIT, ESTEVE**
Titulació:	**ENGINYERIA INFORMÀTICA**
Crèdits:	**30**
Director/Ponent:	**SÀNCHEZ i MARRÈ, MIQUEL**
Codirectora:	**GIBERT OLIVERAS, KARINA**
Departament:	**LSI**

MEMBRES DEL TRIBUNAL *(nom i signatura)*

President:

Vocal:

Secretari:

QUALIFICACIÓ

Qualificació numèrica:
Qualificació descriptiva:

Data:

Gesconda II

Integració de components, redisseny, reimplementació i ampliació de funcionalitats

Alumne: Esteve Olm Margarit

Secretari (Director): Miquel Sànchez i Marrè

President: Ulises Cortés García

Codirectora: Karina Gibert Oliveras

Vocal: Joan Aranda López

Índex

1. Introducció

Les tecnologies de la informació d'avui dia, permeten obtenir grans quantitats de dades a partir de sistemes autònoms que s'encarreguen de mesurar i recol·lectar dades de diferents orígens. Aquestes grans quantitats de dades són poc útils (per no dir gens) si no disposem de mecanismes que transformin aquestes dades en informació. Si a més aquesta informació és útil per alguna finalitat, treure el màxim profit dels processos de captació de dades ens permetrà explicar esdeveniments, controlar comportaments o fins i tot predir-los.

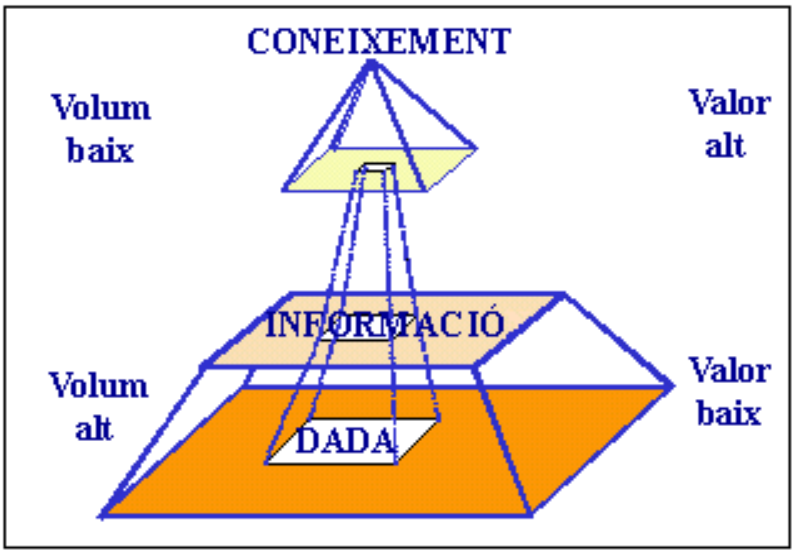


Figura 1.1: Relació entre dada, informació i coneixement (Molina, 1998)

En el moment que podem determinar, que la informació extreta de les dades és útil, i sabem aplicar-la, estem parlant de coneixement.

El nostre projecte (GESCONDA II) s'emmarca dins del camp del descobriment del coneixement i la mineria de dades. Aquest camp necessita eines útils i potents per a efectuar aquesta extracció de coneixement. El projecte té per objectiu la integració dels diversos components, el redisseny, i la implementació de noves funcionalitats, prenent com a punt de partida la versió existent GESCONDA.

1.1. Orígens del Projecte

Gesconda és un sistema d'anàlisi intel·ligent de dades per a la gestió del coneixement de dades medi ambientals (**GES**tió del **CON**eixement de **D**ades **A**mbientals).

El projecte Gesconda neix l'any 2000, amb finançament parcial del govern espanyol (CICyT TIC2000-1011, TIN-2004-1368). La finalitat del projecte és crear una eina de gestió del coneixement implícit de bases de dades medi ambientals. Desenvolupada 100% en Java, aquesta eina sorgeix per la necessitat de tractar la gran quantitat d'informació heterogènia i coneixements implícits que són inherents a les bases de dades resultants de la monitorització dels processos medi ambientals.

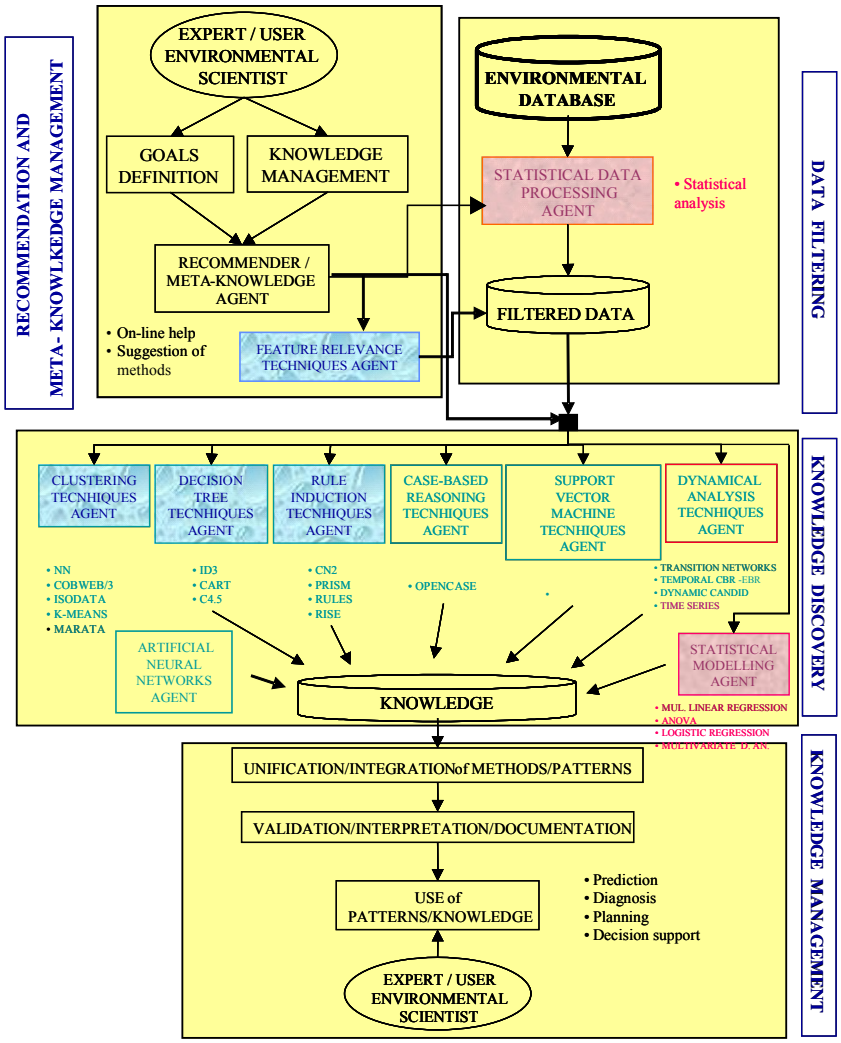


Figura 1.2: Arquitectura de Gesconda

En la figura 1.2 es mostra l'arquitectura inicial del projecte Gesconda, en la que es poden apreciar tres parts ben diferenciades. La primera de tractament i filtrat de dades, una segona capa de descobriment del coneixement i, finalment, una tercera que gestiona el coneixement descobert pels diferents agents.

Dins de la part de descobriment del coneixement, tenim mòduls especialitzats en tècniques de *clustering*, inducció de regles i generació d'arbres de decisió. Cadascun d'aquest mòduls implementa diferents algorismes per aconseguir la seva finalitat.

1.2. Motivacions

Des d'un principi, el projecte Gesconda definia diversos components independents, encarregats de cadascuna de les fases en el procés d'anàlisi, descobriment i predicció de models de dades. Més endavant, es van dur a terme diversos projectes de desenvolupament de software, amb la finalitat d'implementar cadascun d'aquests mòduls i conjuntament fer realitat el projecte global.

Tot i que la finalitat era prou bona, a nivell d'usabilitat tenia algunes mancances que dificultaven el treball diari del usuaris. El fet d'haver estat desenvolupats com a projectes independents, impedia que compartissin dades, essent necessari iniciar diverses aplicacions per a poder executar un procés d'extracció de coneixement de les dades complert. La interfase entre mòduls era a nivell de fitxer de dades amb alguna meta-informació afegida, i no era possible compartir els resultats dels diferents algorismes entre ells.

1.3. Objectius

L'objectiu d'aquest projecte és redissenyar el software Gesconda partint del seu estat actual i proporcionar una nova versió que anomenarem **Gesconda II.** Aquesta nova versió ha de millorar la gestió de les dades, implementar la persistència dels models resultants de l'aplicació dels algorismes i ampliar la pròpia aplicació amb nova funcionalitat. A més, cal que agrupi els diferents components, que actualment

constitueixen Gesconda, en una sola aplicació. Gesconda II ha de disposar d'una interfície gràfica d'usuari unificada, de manera que des d'un mateix entorn es puguin aplicar els diferents processos i algorismes que incorpora la plataforma. Es considera també un objectiu aconseguir que l'aplicació final resultant sigui pràctica, ergonòmica i visualment atractiva.

El primer i clar objectiu del projecte és el redisseny i reimplementació del mòdul de tractament de dades GESP. Aquest mòdul requeria urgentment d'una actualització ja que estava desenvolupat en versions antigues de Java, i a més feia poc ús de les tècniques d'orientació a objectes que ofereix aquesta plataforma. Es preveu des d'un principi que caldrà reescriure aquest mòdul en una bona part, si és que pretenem que s'integri amb la resta.

Un cop tinguem el mòdul Gesp adaptat, caldrà agrupar tots els components de Gesconda en una sola aplicació. Per aconseguir això, primer de tot caldrà unificar els models de dades que utilitzen cadascuna de les aplicacions. Quan s'aconsegueixi aquest objectiu no serà necessari intercanviar la informació entre elles a través de fitxer, amb els inconvenients que això suposava.

Posteriorment es procedirà a unificar la capa de presentació, començant pels menús i acabant amb les diferents vistes d'informació que es requereixen en una aplicació d'aquest estil.

La capa de control de cada mòdul, que inclou les accions a executar i el codi dels diferents algorismes, quedarà separada conceptualment. Els components independents de la versió anterior de Gesconda s'integraran a Gesconda II en una estructura de paquets. La organització en paquets permet a Gesconda II disposar d'una separació física i lògica pel què fa els algorismes segons la seva tipologia, però a la vegada executar qualsevol d'ells des d'una única aplicació.

La unificació dels components de Gesconda en una sola aplicació implica eliminar funcionalitat redundant, que actualment estava replicada en els diferents mòduls i requeria que l'usuari seguís els mateixos processos varies vegades per a un mateix

fi. Exemple clar d'aquesta redundància és la discretització d'atributs (on cada mòdul implementava la seva) o el tractament d'*outliers* i *missings*. Si s'aconsegueix aquesta fita, l'usuari notarà una important millora en la productivitat a l'hora de dur a terme processos de mineria de dades i extracció de coneixement.

Com a últim objectiu es pretén que el software i la documentació generada al llarg d'aquest projecte, serveixi de plataforma i guia per a futurs desenvolupaments de la resta de mòduls i algorismes que poden ser incorporats a Gesconda. Oferint una continuïtat al projecte sense que els mòduls actuals se'n vegin perjudicats per l'evolució de la resta.

2. Context del projecte

2.1. La Mineria de Dades

> *"La filosofia de la Mineria de Dades és la conversió de dades en coneixement per a la presa de decisions. La Mineria de Dades constitueix la fase central del procés d'extracció de coneixement de les bases de dades KDD (Knowledge Discovery in Databases), en aquest sentit la Mineria de Dades és un punt d'encontre de diferents disciplines: l'estadística, 'machine learning', tècniques de bases de dades, sistemes per a la presa de decisions, que juntes, permeten afrontar problemes actuals de les organitzacions pel que fa al tractament de la informació"*

De la definició anterior en podem destacar que el procés de mineria de dades és complex, es composa de moltes disciplines diferents i encara queda molt per explorar en aquest camp. Un dels aspectes més interessants de la mineria de dades és que ens permet extreure informació que prèviament no coneixíem d'un gran volum de dades que a priori no ens diuen res. Per aconseguir això, cal seguir un procés que s'inicia amb la descripció d'un problema a tractar i acaba amb l'aplicació dels resultats generats sobre el aquest.

En el següents figures 2.1 i 2.2, podem apreciar com es defineix el procés de mineria de dades i quina és la metodologia aplicar en el KDD:

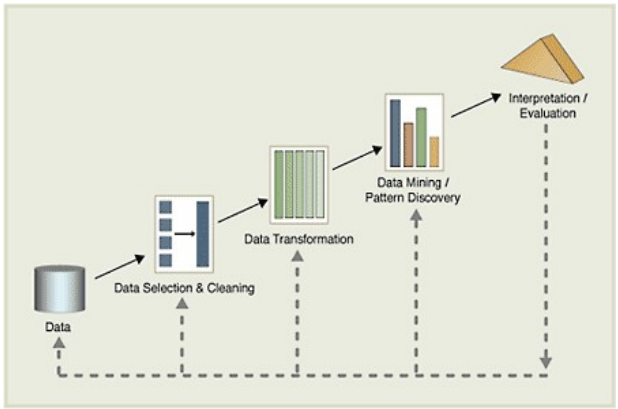


Figura 2.1: Procés iteratiu de KDD

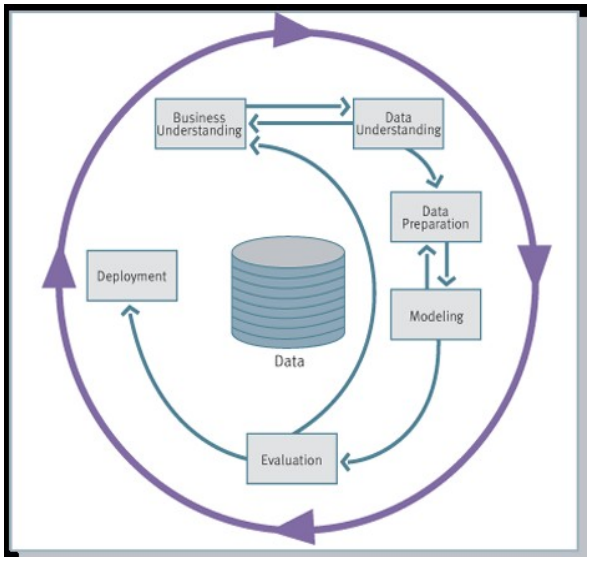


Figura 2.2: Metodologia del procés de descobriment de dades

En el gràfic es poden veure les següents fases:

- *Problem definition*: Definició i comprensió del problema al què es vol trobar solució
- *Data exploration:* Determinació i obtenció de les dades vinculades al problema que es vol resoldre
- *Data preparation:* Filtrat i neteja de les dades per al procés. Es poden derivar noves dades de les existents per tal de facilitar el treball o permetre l'aplicació de més algorismes.
- *Modeling:* Creació de models aplicant tècniques de mineria de dades sobre les dades preparades.
- *Avaluation:* Visualització i validació dels resultats obtinguts en l'etapa anterior. Aquesta fase va molt lligada amb les dues anteriors ja que normalment s'executen de forma iterativa fins a obtenir un resultat coherent i innovador.
- *Deployment:* Aplicació del model descobert i avaluat a noves dades per tal de resoldre el problema inicialment plantejat.

2.2. Estudi del domini

Gesconda és una aplicació que gestiona dades i pretén extreure'n coneixement. De les fases descrites en l'apartat anterior, pretén cobrir les de preparació de les dades, modelatge i avaluació. Aquestes tres fases iteratives són les que podran ser dutes a terme des d'una mateixa aplicació, amb la comoditat que això comporta de cara a l'usuari expert en l'extracció del coneixement.

Partint d'això, i simplificant-ho al màxim, el domini amb què interacciona Gesconda i l'usuari són dades i algorismes. Dels algorismes executats, el que realment interessa, són els models generats o resultats obtinguts.

Les dades provenen de diferents sistemes recol·lectors i per tant poden estar en diferents formats. L'aplicació ha de ser capaç d'interpretar correctament aquestes dades i gestionar-les amb agilitat. Algun dels mòduls anteriors de Gesconda no podia operar amb bases de dades de més de 2000 instàncies. Aquesta no ha de ser una limitació per a la nova aplicació.

Els algorismes no deixen de ser codi Java implementat en els diferents mòduls que composen Gesconda. D'aquests algorismes, ens interessen sobretot els resultats que se'n poden obtenir en funció del seu tipus, ja que en funció d'aquest oferirem diferents representacions visuals a l'usuari.

2.3. Anàlisi d'alternatives

Existeixen al mercat diferents productes que, al igual que Gesconda, pretenen gestionar el coneixement inherent en les dades. En aquest apartat es descriuen els productes alternatius que s'han estudiat en el marc d'aquest projecte. De totes maneres, no s'han considerat com alternativa a la realització de Gesconda II, perquè el que es pretenia des d'un principi, era millorar el posicionament d'un producte ja existent al mercat. D'alguns d'ells però, s'han extret idees de cara a implementar la interfície d'usuari, agafant les que hem considerat adequades i millorant tot el que ha estat possible.

1. Minitab

Minitab és un software dissenyat per a executar funcions estadístiques bàsiques i avançades. Combina l'ergonomia i la facilitat d'ús d'un full de càlcul com Microsoft Excel amb la capacitat d'execució de càlculs estadístics. La versió complerta costa prop dels 1200$, però tot i això és una de les eines més utilitzades en àmbits educatius i professionals. El look&feel i manera d'organitzar continguts han estat font d'inspiració en aquest projecte.

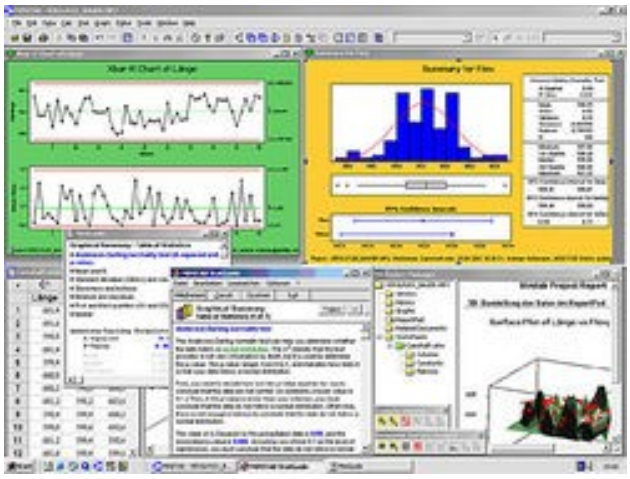


Figura 2.3: Interfície d'usuari de Minitab

2. SPSS

SPSS és un altre programa molt usat per al càlcul estadístic que pren el nom de la companyia desenvolupadora (SPSS Inc.). Creat el 1968 exclusivament per a grans computadores, avui dia té versions per a windows i està molt estès al mercat.

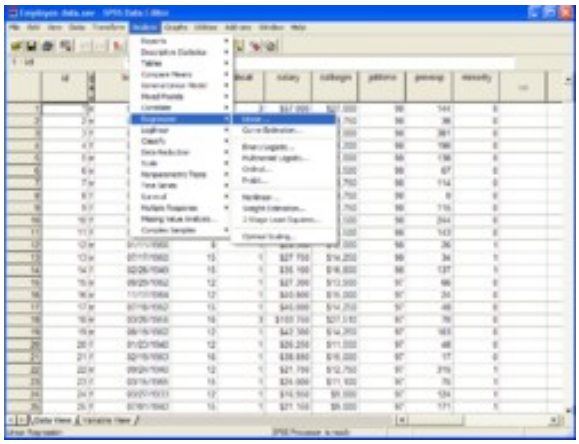


Figura 2.4: Interfície d'usuari de SPSS

3. Synera

Synera és una empresa fundada el 1998 amb seu a Barcelona que té al mercat la versió 4.0 d'una eina de data mining que rep el mateix nom. Synera incorpora en el procés d'adquisició de coneixement tècniques de classificació, xarxes neuronals, arbres de decisió, i clustering entre d'altres. Una dels algorismes de clustering emprat és el Nearest-Neighbour que s'utilitza en aquest cas per tal de predir la classe més propera a una determinada instància.

4. Weka

Aquesta és una eina desenvolupada per Ian H. Witten i Eibe Frank de la universitat de Waikato de Nova Zelanda. Actualment ja està disponible la versió 3.5.6, que incorpora la implementació de diversos algorismes d'aprenentatge automàtic. Està desenvolupada en Java i és gratuïta (open source). Tot i que inicialment no disposava d'una interfície gràfica, la versió actual en té una que està força bé, però no exposa de forma prou clara la funcionalitat que ofereix. Barreja semànticament algorismes en un mateix menú i crea certa confusió a l'usuari. Aquesta eina també

ha estat presa com a referència per a desenvolupar el projecte, amb clara intenció de millorar-la.

5. MineSet 2.0

Programa desenvolupat per Silicon Graphics basat amb una arquitectura client–servidor. Aquest sistema incorpora entre d'altres tècniques de generació de regles d'associació i d'inducció d'arbres de decisió però no treballa amb mètodes d'aprenentatge inductiu no supervisat.

6. Enterprise Miner

Aquesta eina igual que el MineSet 2.0 incorpora tècniques d'aprenentatge com inducció d'arbres de decisió o xarxes neuronals però no disposa d'aprenentatge inductiu no supervisat.

7. R

R és un llenguatge i entorn de treball per al càlcul estadístic i generació de potents gràfics d'anàlisi de dades. És un projecte de GNU similar al llenguatge S, desenvolupat per els Laboratoris Bell (abans AT&T, ara Lucent Technologies). R es pot considerar com una nova implementació de S, però amb l'avantatge de ser software lliure i comptar amb el suport de la comunitat.

R proporciona una àmplia varietat de càlculs estadístics (modelització lineal i no lineal, estadística clàssica, anàlisis de sèries temporals, classificació, ...) i les tècniques gràfiques són altament extensibles.

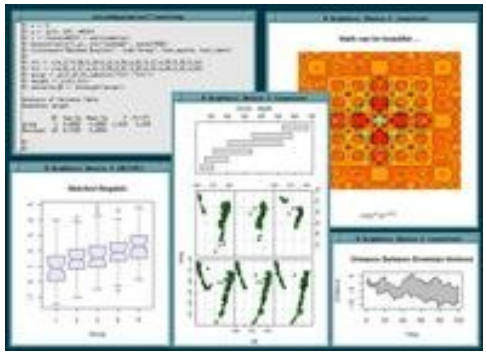


Figura 2.5: Interfície d'usuari de R

2.4. Viabilitat

Com hem comentat abans, Gesconda és un producte actualment en ús i per tant no requereix d'un estudi de mercat per avaluar la seva viabilitat. El que es requereix és una actualització per adaptar-lo als nous temps i facilitar a l'usuari la potència dels seus algorismes. Molts dels productes alternatius que hem vist en l'apartat anterior van començar a implantar-se sense cap interfície d'usuari avançada, funcionant des de línia de comandes. Avui dia, la majoria d'ells incorporen una GUI avançada i per tant Gesconda no pot ser menys i els ha de superar si no vol quedar-se desplaçat.

2.5. Tecnologia i eines de desenvolupament

Com hem explicat en apartats anteriors, Gesconda està desenvolupat 100% en codi Java. Per tant seria molt descabellat utilitzar una altra tecnologia per a realitzar la integració dels seus mòduls. El que sí s'ha procurat és utilitzar les últimes versions estables de Java, per tal d'aprofitar al màxim la potència del llenguatge.

S'ha utilitzat la versió 5 de **Java** ja que la versió 6 (la 7 encara és beta) és relativament massa nova i no està encara prou ben suportada pels entorns de desenvolupament. A més, en la versió Java 6, encara no està disponible la l'Enterprise Edition, fet que fa predir una lenta incorporació al mercat. No es preveu traumàtica la posterior migració a java6, ja que la compatibilitat amb la versió 5 està garantida.

El desenvolupament del codi Java s'ha realitzat sota **Eclipse**. Un potent IDE opensource que permet entre d'altres funcionalitats, navegar de forma fàcil per el codi, compilar, executar i depurar el codi de l'aplicació que s'està desenvolupant i, el més interessant: aplicar potents tècniques de refactoring sobre el codi que permeten treballar de forma ràpida i desordenada. Un cop el codi és funcional, aquestes tècniques permeten netejar, ordenar i ajustar el codi als estàndards de desenvolupament establerts al projecte, generant un producte de qualitat i ben documentat.

El codi desenvolupat s'ha emmagatzemat en un repositori **SubVersion** que permet un potent control de versions en treball local i remot. A més permet a diferents equips (no ha estat el cas) compartir el codi desenvolupat sense interferir entre ells. *SubVersion* és un producte de *tigris.org* sota llicència Apache/BSD, utilitzat cada cop més en substitució del conegut CVS.

Per tal d'adaptar el codi dels diferents mòduls de Gesconda (codi duplicat, però amb certes modificacions) s'ha utilitzat una eina per a la comparació de fitxers i directoris. En aquest cas, l'eina escollida ha estat **Meld**, amb llicència GPL. Amb aquesta eina s'han comparat els directoris del codi font i els propis fitxers per tal d'eliminar classes

redundants i incorporat tot el codi comú en un *package* compartit entre els diferents mòduls.

Al desenvolupar la funcionalitat de la capa de presentació, es requeria d'una millora en les classes que generaven els gràfics d'anàlisi (histogrames, gràfics de barres, plots, etc.) i la implementació actual era força limitada a l'ús que tenia. S'ha incorporat una llibreria LGPL anomenada **jfreechart**, que permet generar gràfics més potents i adaptables a les necessitats de l'aplicació. *JFreeChart*, a la vegada fa servir la llibreria *jcommons*, també LGPL i també ha estat incorporada al codi.

El producte final ha estat empaquetat en un sol arxiu entregable, utilitzant **one-jar** (*http://one-jar.sourceforge.net*), que permet incloure llibreries en un mateix arxiu jar reemplaçant el ClassLoader de la màquina virtual per un especialitzat que és capaç de descomprimir el projecte en temps d'execució.

2.6. Metodologia

La metodologia emprada per a dur a terme el desenvolupament d'aquest projecte, s'emmarca dins de la tipologia de metodologies àgils de desenvolupament. Dins d'aquesta tipologia, hi trobem exemples ben coneguts com podria ser la programació extrema (coneguda com XP d'*eXtremme Programming*), però la que hem utilitzat al llarg de l'etapa de desenvolupament d'aquest projecte és la que es coneix amb el nom d'**Scrum**.

Scrum és una simplificació d'XP, que permet dur a terme desenvolupaments complexos en un temps breu, amb excel·lents resultats i sense la càrrega documental que requereixen altres metodologies més serioses. *Scrum* és el terme anglès que s'utilitza per denominar a una melé en Rugby, i certament, té alguna similitud amb aquest esport.

La metodologia escollida, però, no pot ser aplicada a qualsevol tipus de projecte. Cal tenir en compte algun dels seus inconvenients com per exemple, que l'equip que es fa càrrec del desenvolupament ha de ser prou expert per no dependre de la

documentació generada per personal situat en nivells jeràrquicament superiors, ni tenen la possibilitat de delegar feines més rutinàries a nivells inferiors.

En el marc d'aquest projecte, l'aplicació d'aquesta metodologia és prou encertada, ja que no es disposa d'un equip amb diferents perfils per a la realització de les tasques, i els formalismes de la documentació de les fases de desenvolupament no són molt estrictes. Gesconda no és un producte crític: no gestiona dades vitals per a una empresa, ni s'aplica en entorns que requereixin altes mesures de seguretat i encriptació, ni és un mecanisme de control de maquinària perillosa o de la que en depenguin la vida de les persones. En projectes de missió crítica no és viable l'aplicació de metodologies àgils per al desenvolupament.

A la pàgina següent, es pot apreciar de manera molt resumida, com està definida la metodologia Scrum, aplicada en aquest projecte.

Cal remarcar, que per aquest projecte, tant l'Scrum manager, com l'equip de treball són una mateixa persona (l'autor d'aquesta memòria i del mateix projecte) i com a propietaris del producte o persones interessades tindríem el Director i CoDirectora del projecte.

Les reunions diàries de l'Scrum manager amb l'equip de treball no han estat necessàries (evident, al ser la mateixa persona), però sí s'han realitzat setmanalment les presentacions d'increments, suggerències i anuncis de propers sprints amb els propietaris del producte.

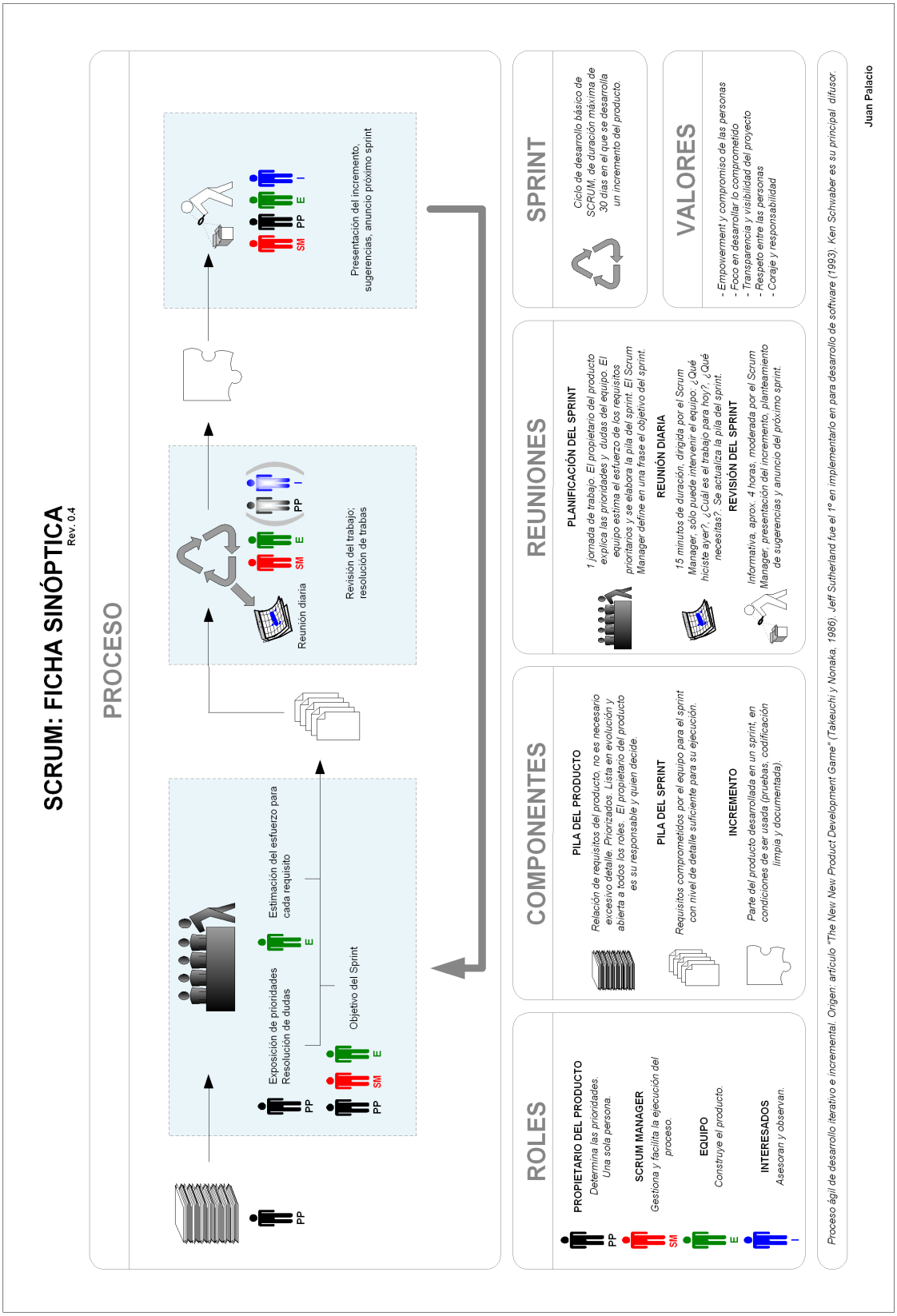


Figura 2.6: Fitxa sinòptica de la metodologia Scrum (Wikipedia)

3. Modelització del domini

3.1. Particularització del domini

Entenem com a Bases de Dades a fitxers plans en diversos formats, que contenen una sèrie d'instàncies o observacions. Cadascuna d'aquestes instàncies conté valors corresponents a un conjunt d'atributs (el conjunt d'atributs és el mateix per a totes les instàncies). En una situació idíl·lica, totes les instàncies tindran valors per a tots els atributs, però sabem d'entrada que això no sempre és així (de fet gairebé mai) i que cal interpretar i tractar els valors que manquen (*missing*).

Els valors que poden tenir les instàncies són sempre números reals o cadenes de text. Gesconda no tracta amb cap altre tipus de dades que no sigui aquests. Dels atributs que defineixen les dades necessitem saber-ne el tipus, ja que en funció d'aquest podrem aplicar diferents algorismes per calcular distàncies o regles. Diferenciarem entre atributs amb valors continus (que anomenarem **variables quantitatives**) i atributs amb valors discrets (**variables qualitatives**). D'aquestes últimes en diferenciarem també dos tipus: les qualitatives ordenades i les qualitatives no ordenades. L'aplicació dels algorismes sobre aquests tipus de valors varia en funció de la seva naturalesa.

Els resultats dels algorismes són també part del domini a ser tractat per l'aplicació. Els algorismes de classificació generen noves variables qualitatives que permeten visualitzar la base de dades en prototips. Caldrà només modelitzar aquests prototips com a agrupacions d'instàncies de la base de dades en funció de la variable qualitativa a la que pertanyen.

Els algorismes d'inducció de regles generen conjunts de regles que posteriorment poden ser aplicades a la base de dades. Hem de modelitzar aquestes regles en el domini per tal d'assegurar-ne el correcte tractament i persistència. Necessitarem poder importar i exportar conjunts de regles, per tal de poder-les executar sobre la base de dades o bé utilitzar regles generades en altres sistemes (formats CLIPS)

Els algorismes d'arbres de decisió generen arbres. Cada arbre es modelitza com un conjunt de nodes amb una relació pare-fill i el node emmagatzema informació sobre els atributs i condicions que s'apliquen en ell. Els arbres de decisió, es desaran també amb el model de dades per a la posterior gestió des de l'aplicació entre diferents execucions del software.

Els algorismes que són capaços d'executar els diferents mòduls de Gesconda, així com les funcions de càlcul de distàncies, també són objectes del domini, però no cal dedicar-los especial atenció. Per una banda perquè ja estan implementats i només cal assegurar-se que interaccionen de forma correcta amb la resta del domini unificat de l'aplicació. Per altra banda, perquè estan incorporats en el codi i no cal gestionar-ne la persistència ni l'entrada/sortida.

3.2. El punt de partida

En aquest apartat es descriuen cadascun dels components que fins a la data de realització del projecte han format part de Gesconda i es detalla per a cadascun d'ells la seva funcionalitat. Aquests components han estat el punt de partida al desenvolupament del nou Gesconda, millorant-los en tot el que ha estat possible i integrant-los en una sola aplicació.

3.2.1. Gesp v1.1

Gesp és el mòdul de preprocés de les dades abans de l'aplicació dels algorismes per a l'extracció de coneixement. Fou desenvolupat en Java per estudiants d'estadística sense grans coneixements de programació, fet que cal destacar com a mèrit degut a què és prou funcional i bastant avançat al seu temps.

D'entre la seva funcionalitat podem destacar que permetia manipular la base de dades, crear nous atributs basats en variables estadístiques, realitzar diferents tipus d'anàlisi estadística descriptiva, gràfics i calcular models estadístics predictius com la regressió lineal o l'anàlisi de la variància.

En les següents imatges podem veure algunes captures de pantalla de l'aplicació Gesp, mòdul de preprocés de dades del software Gesconda:

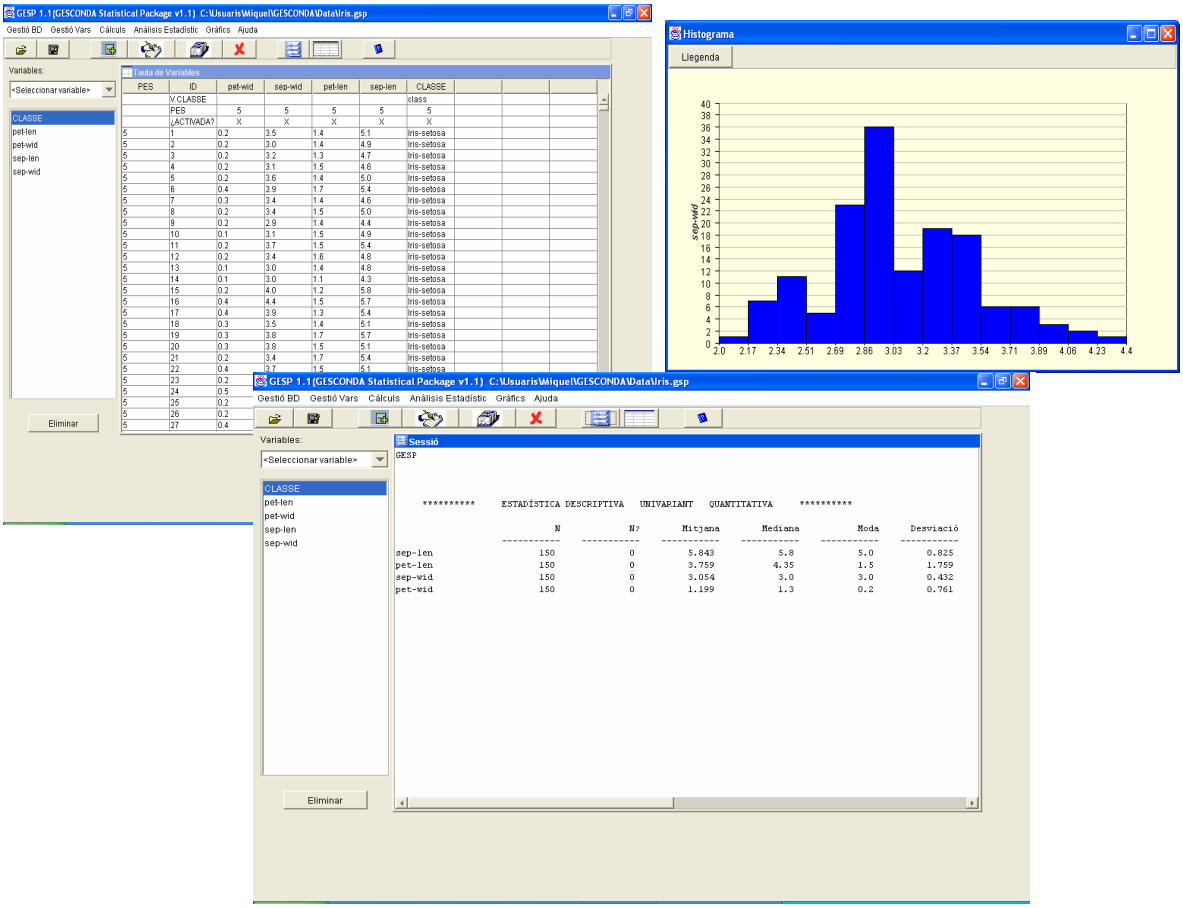


Figura 3.1: Captures de pantalles de Gesp1.1

Gesp, però, oferia algunes mancances importants que calia millorar i han estat objectius definits d'aquest projecte des d'un principi. Les mancances funcionals més destacables del mòdul Gesp eren:

- Dificultat per manipular dades directament sobre la matriu
- Impossibilitat de treballar amb bases de dades de més de 2000 registres
- Incompatibilitat parcialment resolta amb el format de fitxer utilitzat per a la comunicació amb la resta de mòduls

A més d'això, requereix d'actualització de la interfície gràfica d'usuari.

3.2.2. Clustering

Aquest és el mòdul encarregat de l'execució d'algorismes de descobriment de prototipus i classificació de les dades. A diferència de l'anterior, incorpora una arquitectura molt més ben definida i una programació orientada a objectes molt ben estructurada. Fa un ús intensiu però no excessiu de l'herència i proporciona classes d'ajuda per a la generació de vistes programàticament. Segueix un patró MVC (model-vista-controlador) i el codi està bastant ben organitzat sota aquest paradigma. El mòdul de Clustering executa els algorismes de K-Means, Nearest-Neibourghs, Isodata, Marata i CobWeb/3

En les següents imatges es mostren algunes de les pantalles de l'aplicació de Clustering:

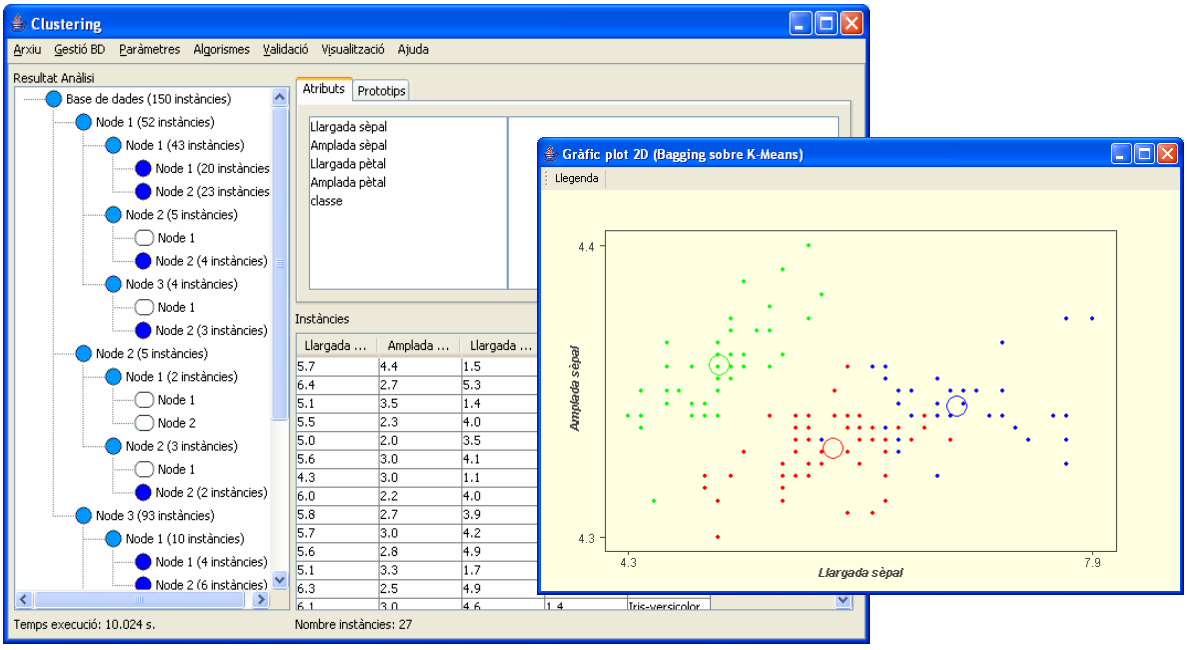


Figura 3.2: Captures de pantalla del mòdul de Clustering

Com hem comentat anteriorment, Clustering és un producte ben desenvolupat i té poques deficiències tècniques i funcionals. El fet que ens motiva a integrar-lo i redissenyar aquest mòdul en aquest projecte és l'àmbit d'actuació, que queda limitat a l'aplicació d'un sol tipus d'algorismes. Com hem comentat en els apartats anteriors, l'objectiu del projecte és unificar els subprocessos d'extracció de coneixement en una sola aplicació.

De la funcionalitat millorable a Clustering, destaquem la impossibilitat de desar els models generats ni d'aplicar validacions més que sobre l'últim algorisme executat, fet que requeria tornar a executar els mateixos algorismes n vegades fins a poder obtenir un resultat coherentment validat. A més, el fet de no disposar de funcionalitat de preprocés incorporada, feia que fos necessari dependre d'altres aplicacions (Gesp) per a la gestió inicial de les dades. Degut a què el procés d'extracció de coneixement és iteratiu, calia passar per fitxer una i altra vegada com a única interfase de comunicació entre els components de Gesconda.

3.2.3. Inducció de Regles i Feature Weighting

Aquest mòdul s'encarrega de l'execució dels algorismes de Rules, Prism, CN2 i Rise, a més d'incorporar algorismes especialitzats en determinar la rellevància dels atributs en la base de dades. Fou desenvolupat amb posterioritat al mòdul de Clustering de manera independent, prenent com a punt de partida el codi d'aquest modul, adaptant-lo i evolucionant-lo a les necessitats de Rules. Això ha provocat que es disposi del mateix model de dades duplicat però amb certes diferències importants per a la correcta execució dels algorismes. Un dels objectius importants d'aquest projecte és precisament eliminar aquesta duplicitat i integrar els models per a què tots els algorismes funcionin sobre les mateixes dades correctament.

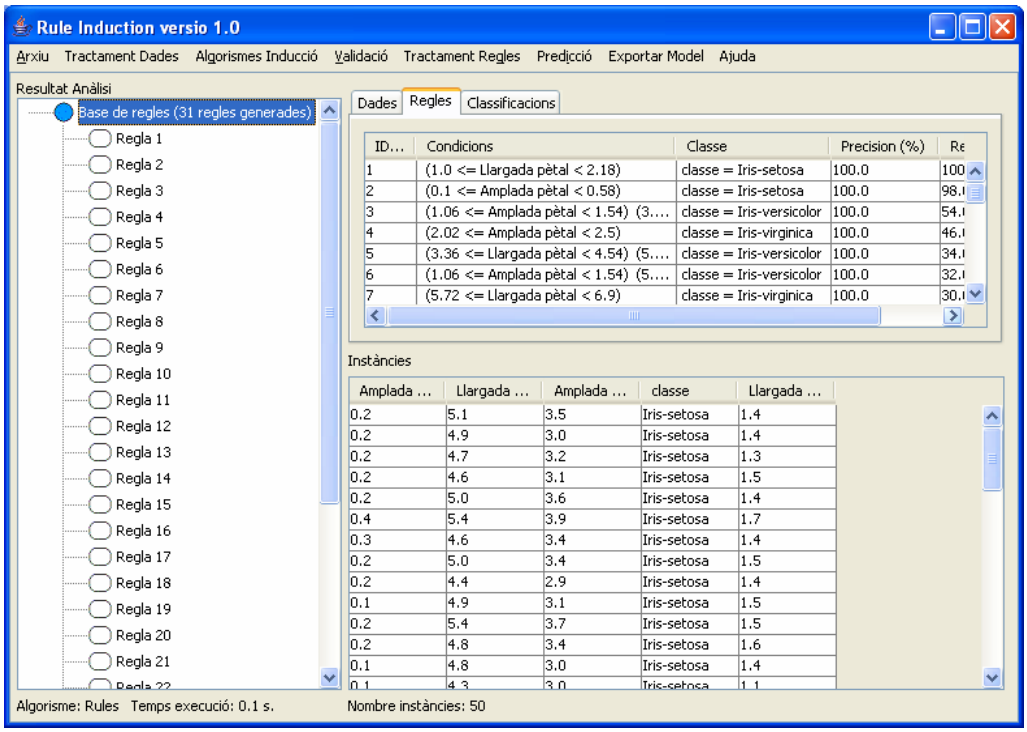
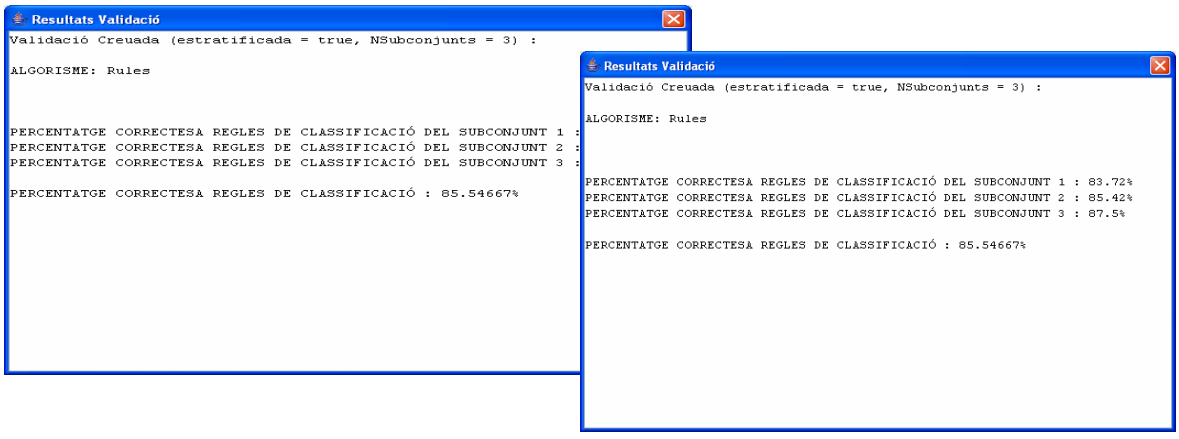


Figura 3.3: Captures de pantalla de Regles de Decisió i Feature Weighting

En les anteriors imatges es poden veure algunes de les pantalles de l'aplicació de regles de decisió.



Es pot apreciar certa similitud amb el mòdul de Clustering, fet que fa pensar que seria interessant agrupar aquests components en un de sol, i motiva la necessitat d'aquest projecte.

3.2.4. Decision Tree

La generació d'arbres de decisió és en Gesconda un procés bastant autònom de la resta de components. Fou desenvolupat per un estudiant d'Erasmus i fa poc ús dels components desenvolupats prèviament. Incorpora el seu propi model de dades i no fa distinció d'atributs com els mòduls anteriors. També trenca l'estructura de les pantalles i interfícies d'usuari que defineixen els seus predecessors. Tot i això, l'aplicació és prou correcta, al ser capaç d'executar algorismes de generació d'arbres de decisió com el ID3, el C4.5 i el CART, a més d'incorporar algorismes de *post-prunning pessimístic* i *cost-complexity* i validació de resultats obtinguts.

En les següents captures de pantalla es pot veure l'aplicació d'arbres de decisió del software Gesconda:

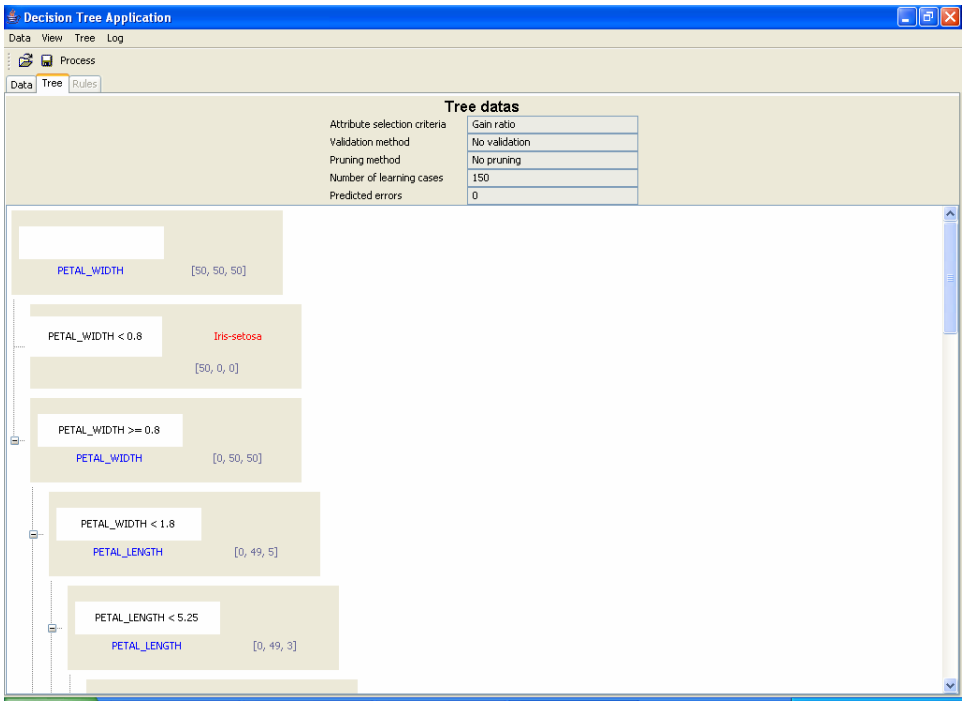


Figura 3.4: Captures de pantalla de Decision Tree

Com es pot apreciar, la representació gràfica dels arbres generats és poc natural, i tot i que amb posterioritat al projecte es van fer algunes millores, tampoc són satisfactòries per als requeriments de l'usuari. L'objectiu del projecte pel què fa a aquest mòdul és integrar també el model de dades i millorar notòriament la

representació gràfica dels arbres generats. A més de poder emmagatzemar i gestionar els arbres generats sense haver de tornar a executar els algorismes en cada ocasió.

Si es requereix més informació d'aquests mòduls sobre com van ser implementats i quina funcionalitat cobreixen amb major detall, a la bibliografia hi trobareu les referències de les memòries dels projectes realitzades pels seus autors.

4. Disseny de l'aplicació informàtica

4.1. Requeriments

L'anàlisi de requeriments pren com a punt de partida els components desenvolupats de Gesconda amb les seves mancances i limitacions. Més enllà s'han incorporat nous requeriments que permetin a Gesconda II evolucionar i ser un producte destacat, al mateix nivell que els productes comercials existents al mercat. Els requeriments han estat agrupats en:

- Requeriments tecnològics
- Integració dels mòduls existents
- Millora de la interfície gràfica d'usuari
- Noves funcionalitats
- Facilitat de manteniment

4.1.1. Requeriments Tecnològics

Req.1.1. *Adaptació a Java5*

Eliminar l'ús de mètodes *deprecated* en versions anteriors de Java. Tot i que no comporten errors en l'execució, alguns d'ells fan que el repintat de pantalles no sigui correcte i l'aplicació vagi més lenta del que podria anar.

4.1.2. Requeriments d'integració

Req.1.1. *Integrar en la nova versió de Gesconda unificada, tots els mòduls existents*

Els mòduls que actualment formen part de Gesconda són:

- Clustering i Bagging
- Inducció de regles i Feature Weighting
- Inducció d'arbres de decisió i algorismes de post-prunning.
- GESP: Mòdul d'anàlisi estadística

Req.1.2. *Unificació del model de dades*

Els mòduls que composen Gesconda fan servir models de dades similars però no idèntics. Cal que utilitzin el mateix per tal que puguin compartir-lo per a què puguin ser incorporats en una sola aplicació.

Req.1.3. *Unificació de la interfície d'usuari*

Permetre a tots els mòduls de Gesconda executar-se dins un mateix marc o finestra, compartint les vistes que mostren elements del model unificat, i integrant totes les accions en un mateix menú.

Req.1.4. *Eliminació de funcionalitat redundant*

Al ser Gesconda una aplicació formada per components independents, molta de la funcionalitat necessària per a realitzar un anàlisi ha estat replicada en cadascun d'ells. Per exemple la selecció i càlcul de distàncies, la discretització d'atributs, la gestió d'atributs actius, tractament d'*outliers*, *missings* i definició de l'atribut classe. Tota aquesta funcionalitat ha de ser unificada i centralitzada en un sol punt.

Req.1.5. *Adaptació del codi de GESP*

Aquest mòdul de tractament previ de dades és el que requereix més dedicació,

ja que no utilitza tècniques d'orientació a objectes i el seu model de dades té molt poc en comú amb la resta.

4.1.3. Requeriments de millora de la interfície gràfica d'usuari

Req.1.1. *Interfície gràfica ergonòmica*

L'usuari ha de disposar, a més dels menús habituals a la part superior, de barres d'eines amb les operacions més comunes en cada vista i de menús contextuals associats als elements que s'hi mostren. Per tal d'estalviar temps en la navegació del menú global de l'aplicació.

Req.1.2. *Visualització a la interfície de diferents paràmetres del sistema*

Cal mostrar la base de dades activa, la funció de distància activa, l'atribut classe, l'algorisme en execució i la grandària de la matriu de dades

Req.1.3. *Exportació de gràfics*

Els gràfics d'anàlisi de dades que genera l'aplicació han de ser exportables a algun format per tal de poder ser utilitzats en documents o altres aplicacions.

Req.1.4. *Distinció visual de valors outliers i missings a la matriu de dades*

De cara a facilitar el tractament de les dades previ a l'aplicació d'algorismes de mineria de dades, cal que els valors estranys en la base de dades siguin fàcilment localitzables per l'usuari i pugui fer-ne un tractament.

Req.1.5. *Millorar la funcionalitat dels gràfics d'anàlisi de dades*

Els gràfics que genera l'aplicació són vistosos i correctes, no obstant, no es poden exportar (excepte si es fa una captura de pantalla) i tampoc variar-ne el tamany ni les propietats. Les mesures que apareixen en els eixos a vegades se superposen per la naturalesa dels valors, fent que siguin il·legibles per l'usuari.

Req.1.6. Representació gràfica dels arbres de decisió

L'actual implementació de la visualització d'arbres de decisió, tot i ser correcta, no permet fer-se una idea de la representació del model. Cal desenvolupar una vista més adequada a la realitat conceptual dels arbres de decisió on la jerarquia entre nodes sigui més evident i les etiquetes dels enllaços siguin més clares.

4.1.4. Noves funcionalitats

Req.1.1. Importació i exportació de dades en diferents formats

Cal que Gesconda II sigui capaç de llegir dades en formats habitualment utilitzats en aplicacions de gestió i realitzi les conversions necessàries per a poder interpretar i manipular fitxers generats per diferents programes.

Req.1.2. Permetre l navegació i edició de valors sobre la matriu de dades

Els mòduls d'anàlisi no permeten modificar les dades perquè pressuposen que el mòdul GESP de tractament de dades ho ha fet prèviament. El mòdul GESP no és gaire ergonòmic en aquest aspecte i no permet modificar les dades sobre la vista com si es tractés d'un full de càlcul. Cal implementar aquesta interacció amb la vista per a facilitar aquesta operativa.

Req.1.3. Possibilitat de fer copy&paste dels valors de la matriu de dades amb altres aplicacions que gestionin matrius de dades

Permetre que, per exemple, des de MS Excel es puguin copiar files o columnes i enganxar-les a la matriu de dades de Gesconda. També ha de ser possible realitzar la operació inversa: copiar dades de Gesconda i enganxar-les a Excel.

Req.1.4. Permetre gestionar més de 2000 instàncies en una Base de Dades

La implementació actual del mòdul de tractament de dades no permet obrir ni tractar bases de dades amb més de 2000 instàncies. Utilitzar memòria

dinàmica per a què sigui la limitació física de la màquina on s'executa qui impedeixi obrir bases de dades grans i no pas el codi de l'aplicació.

Req.1.5. *Persistència dels resultats de l'aplicació*

L'usuari ha de poder desar el treball fet i poder restaurar-lo amb posterioritat, entenent com a resultats de l'aplicació tan les noves dades generades com els models obtinguts de l'aplicació dels algorismes. Actualment només es permet desar la base de dades amb les instàncies, però cal també que els resultats dels algorismes executats siguin guardats per al posterior anàlisi i validació.

Req.1.6. *Superposició de TS-Plot de varies variables*

Permetre la generació de gràfics TS-Plot superposats per tal de poder comparar variables que segueixin sèries temporals en un sol gràfic.

Req.1.7. *Disposar d'un sistema d'ajuda a l'usuari*

L'aplicació ha d'incorporar una ajuda per a l'usuari que en tot moment pugui consultar la informació dels algorismes a executar.

4.1.5. Requeriments de facilitat de manteniment

Req.1.1. *Facilitar la modificació dels menús*

El fet d'integrar els mòduls en un sol marc, implica disposar d'un menú unificat amb totes les accions disponibles a Gesconda II. Aquest menú ha de ser fàcilment configurable per tal d'adaptar-lo a l'usuari si és necessari i en cas de voler incorporar o variar la funcionalitat dels algorismes existents.

Req.1.2. *Suport multi-idioma*

Permetre la generació de menús, pantalles, diàlegs i missatges d'error en diferents idiomes en funció de les preferències de l'usuari.

Req.1.3. *Generar documentació adequada per al posterior manteniment de l'aplicació*

Cal que les tècniques aplicades en el desenvolupament de l'aplicació siguin de domini públic per a posteriors persones que es facin càrrec del manteniment evolutiu i correctiu de Gesconda II. Cal marcar i documentar unes pautes de treball i patrons de disseny a utilitzar en cas de voler modificar el codi de l'aplicació.

4.2. Disseny conceptual

L'aplicació Gesconda II ha estat desenvolupada, al igual que els seus predecessors, seguin el paradigma Model-Vista-Controlador. Podrem veure també que a més baix nivell s'utilitzen altres tècniques per gestionar events, actualitzacions i repintats de pantalla, però tots aquests mecanismes són intrínsecs de la tecnologia escollida (Java swing) i poc tenen a veure amb el disseny de l'arquitectura de l'aplicació. Els considerarem detalls d'implementació, tot i que en veurem algun exemple.

4.2.1. Model

La integració del model de dades a Gesconda II s'ha aconseguit fusionant els arbres de classes dels mòduls de Clustering i Regles. Donada la seva similitud, amb l'ajuda de programes de control de versions i de comparació visual de diferències, s'aconsegueix obtenir una primera versió compilable i funcional del model. Després s'integren la resta de mòduls amb el model únic per a què utilitzin els mateixos components.

Al implementar la integració s'havia de garantir dues fites importants. Per una banda, garantir que totes les aplicacions fan servir una única instància del model de dades i que no es creen rèpliques en memòria de les dades en cap punt de l'aplicació. D'altra banda, les aplicacions que no feien servir inicialment un model amb aquesta estructura, haurien de tenir fàcil accés al model sense haver de replantejar la seqüència de crides per realitzar la immersió del paràmetre model. El patró de disseny que més satisfà aquests requeriments és el que es coneix amb el nom de *Singleton.*

El patró *Singleton* és sovint discutit perquè es pot interpretar com un eufemisme per a declarar una variable global. En aquest cas, però, està justificat per la necessitat de fer accessible el model de dades global a qualsevol algorisme de Gesconda, actuant la classe CMModel com a interfície front-end o façana de la resta de classes del model de dades que sí disposen de mètodes accessors especialitzats.

En el següent diagrama es mostren al nivell més alt les classes del model de Gesconda:

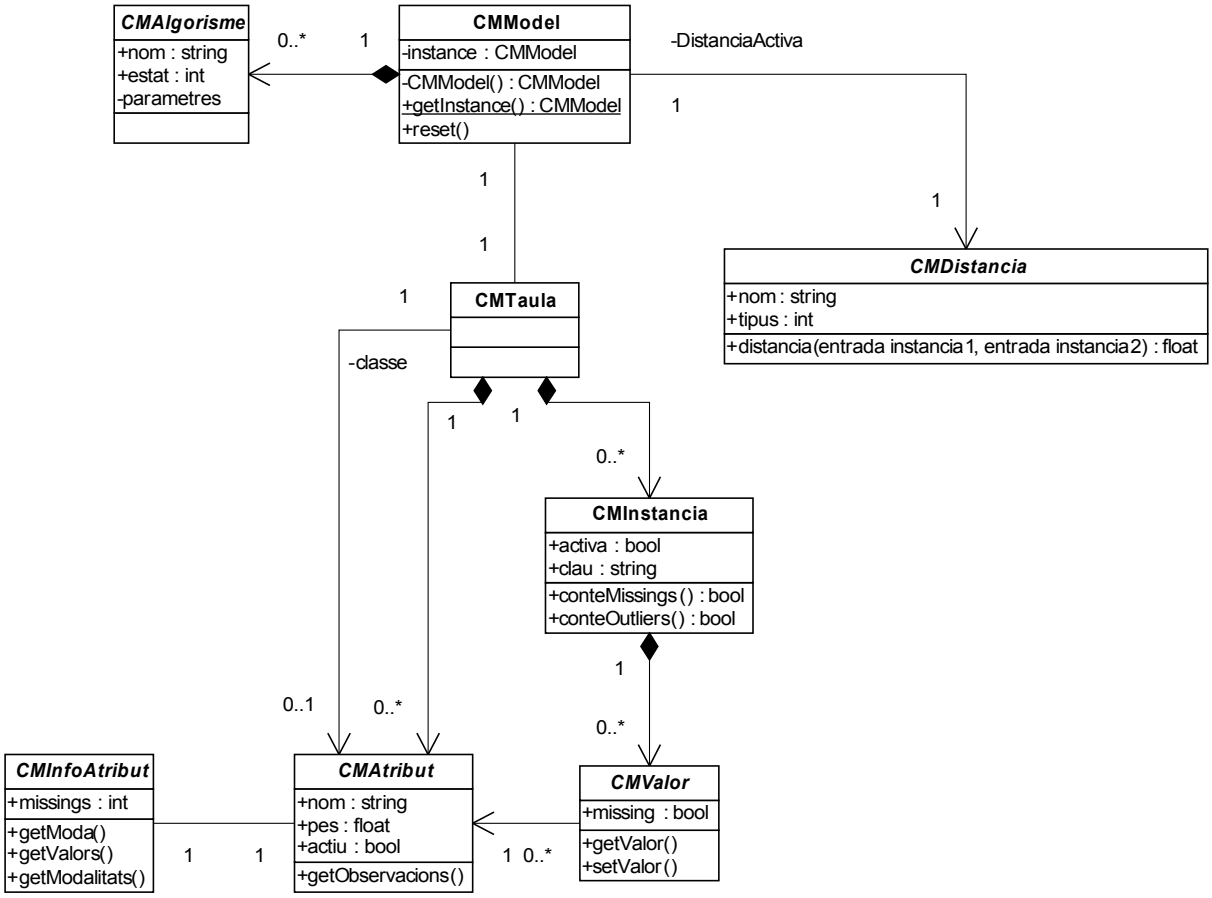


Figura 4.1: Diagrama de classes del model de l'aplicació

Totes les classes del model segueixen una nomenclatura estandaritzada, i és per això que comencen per CM (Classe Model).

Del diagrama destaquem la classe CMModel amb el seu mètode estàtic getInstance() que implementa el *Singleton*, i les relacions que té amb les altres classes.

A grans trets, Gesconda disposa d'un model de dades que inclou una taula o matriu de dades (CMTaula). Aquesta taula conté una llista d'instàncies (CMInstancia) i una llista d'atributs (CMAtribut), a més de pode especificar quin d'aquests atributs és l'atribut classe.

Cada instància conté un conjunt de valors (CMValor) i cadascun d'aquests valors està lligat a un dels atributs de la taula. La navegabilitat de les relacions és complerta en el codi, però en el diagrama només s'ha especificat les més habituals.

Cada atribut disposa d'una informació addicional que s'ha separat per raons semàntiques en la classe CMInfoAtribut. Aquesta part del model és abstracta, i en funció dels tipus d'atributs tenim subclasses que en defineixen el comportament.

Per últim, a la part dreta del diagrama principal, podem apreciar la distància com a element del model. Tot i que no és en realitat cap objecte del domini que calgui gestionar el seu estat, Gesconda tipifica els diferents tipus de funcions distància que pot aplicar entre instàncies de la base de dades amb aquestes classes. Cada una d'elles implementa l'algorisme de càlcul de la distància i és utilitzat per els algorismes que realitzen càlculs. El model global en té sempre una de seleccionada que és la que s'utilitza com a funció de distància activa.

A continuació es mostra el diagrama de classes corresponent a l'especificació de la classe abstracta CMAtribut i corresponents:

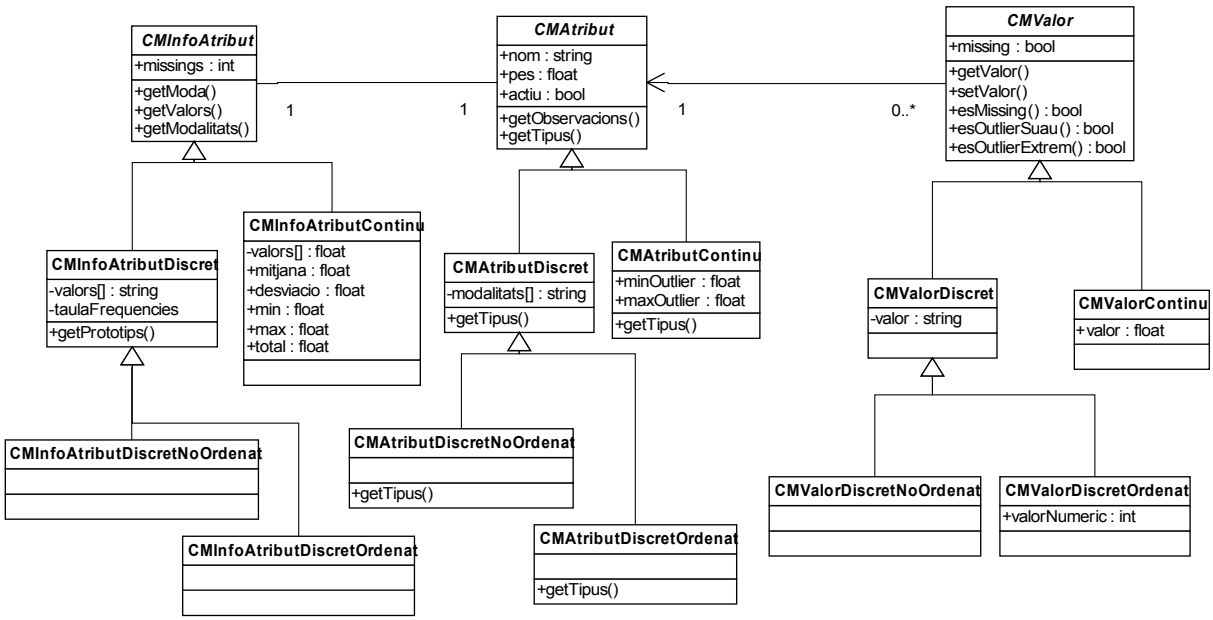


Figura 4.2: Diagrama de classes de l'especialització de tipus d'atributs

S'hi poden observar les generalitzacions i especificacions dels tipus d'atribut, en relació amb els tipus detectats en l'apartat de Modelització del domini. Algunes de les classes gairebé no aporten funcionalitat (mètodes ni atributs) però són d'utilitat per tal de diferenciar el tipus d'atribut que estem tractant. A més permet que el model sigui extensible si es vol afegir nous tipus d'atributs o bé especialitzar encara més els que disposa.

A més dels atributs, instàncies i valors. El model també disposa d'una col·lecció d'algorismes que es corresponen amb les execucions que s'han anat fent en la sessió de treball. Cada algorisme és d'un tipus determinat (en relació amb els diferents mòduls de Gesconda) i en conseqüència ofereix diferents models obtinguts per a ser estudiats. En el següent diagrama es mostren els diferents tipus d'algorismes que gestiona Gesconda i la informació associada que pot ser mostrada de cadascun d'ells:

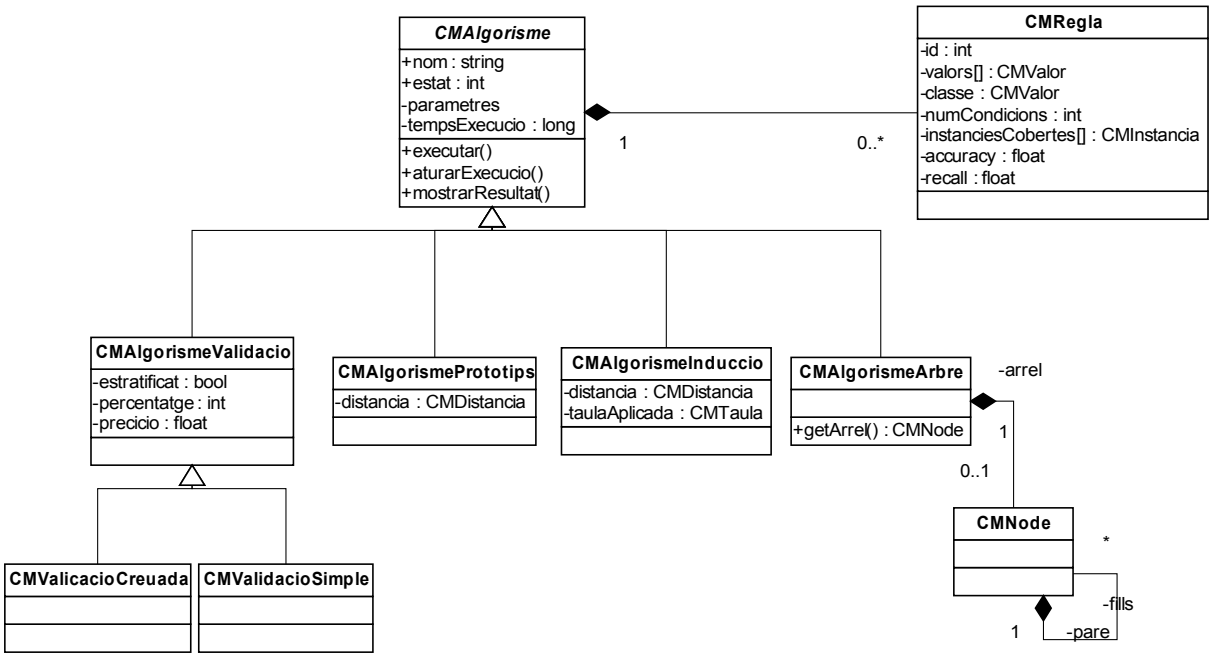


Figura 4.3: Classes que representen els diferents tipus d'atributs

Com es pot apreciar en el diagrama, la classe algorisme i especialitzacions, implementen el patró plantilla, a més que permeten al controlador, com veurem més endavant, gestionar l'estat i l'execució de forma unificada.

En les següents figures es detallen les classes que hereten de cada tipus d'algorisme, encarregades d'implementar i gestionar els algorismes concrets que Gesconda és capaç d'executar:

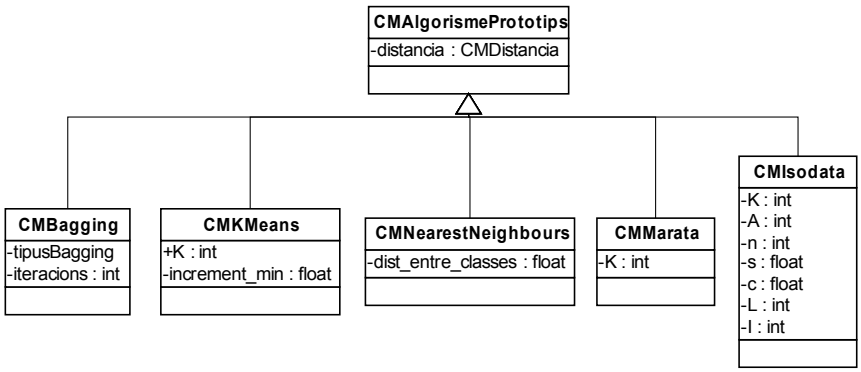


Figura 4.4: Algorismes de prototips

Els algorismes de prototips són els corresponents al mòdul de clustering.

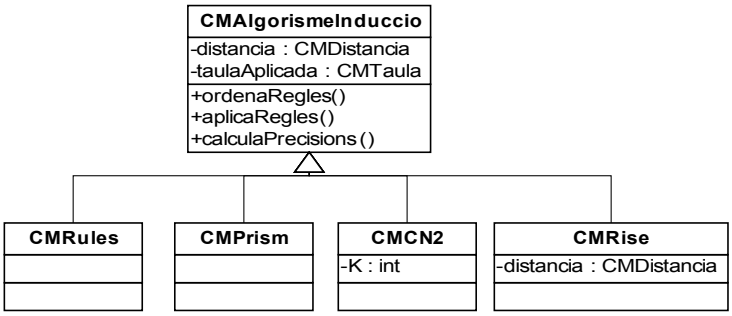


Figura 4.5: Algorismes d'inducció de regles

Els algorismes d'inducció es corresponen amb l'anterior mòdul de regles, i a més de la funcionalitat genèrica d'un algorisme, aporten la funcionalitat de gestió i aplicació de les regles d'inducció.

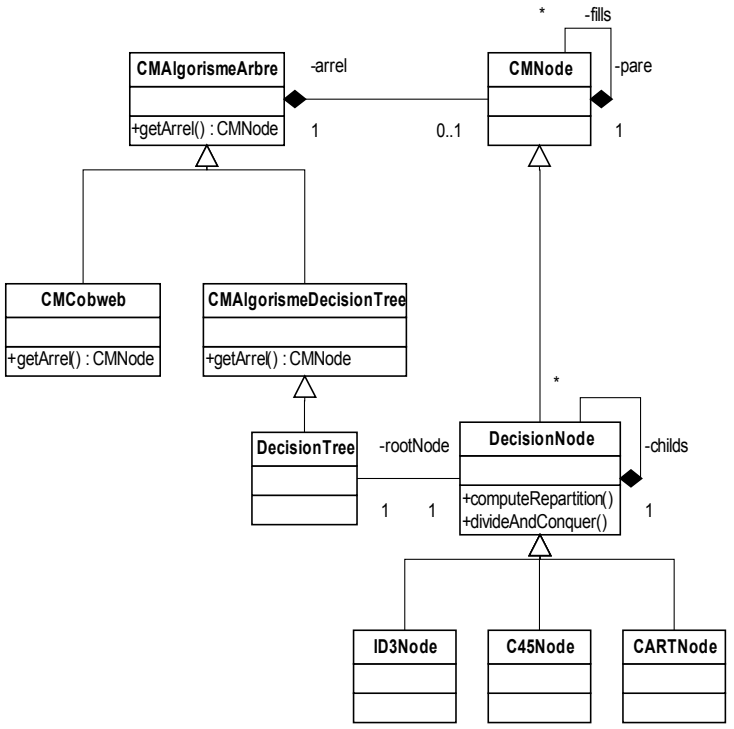


Figura 4.6: Algorismes d'arbres de decisió

En algorismes de tipus arbre s'hi inclou el CobWeb, corresponent a clustering, i tota la generació d'arbres de l'anterior mòdul de Decision Tree. S'ha redefinit una herència entre CMNode i DecisionNode per a poder gestionar els arbres generats pels dos tipus d'algorismes de forma unificada. Els arbres de decisió es generen de dalt a baix, mentre que l'arbre de CobWeb es genera de baix a dalt, però gràcies a l'herència això no és cap inconvenient.

4.2.2. Vista

En aquest apartat es mostren els diagrames de classes corresponents als components que actuen de vista i aporten la funcionalitat de interfície gràfica d'usuari. Pràcticament tota aquesta part ha estat implementada de nou, perquè calia proporcionar una funcionalitat més genèrica a tots els components de la plataforma Gesconda.

El diagrama de classes de la capa de presentació és aquest:

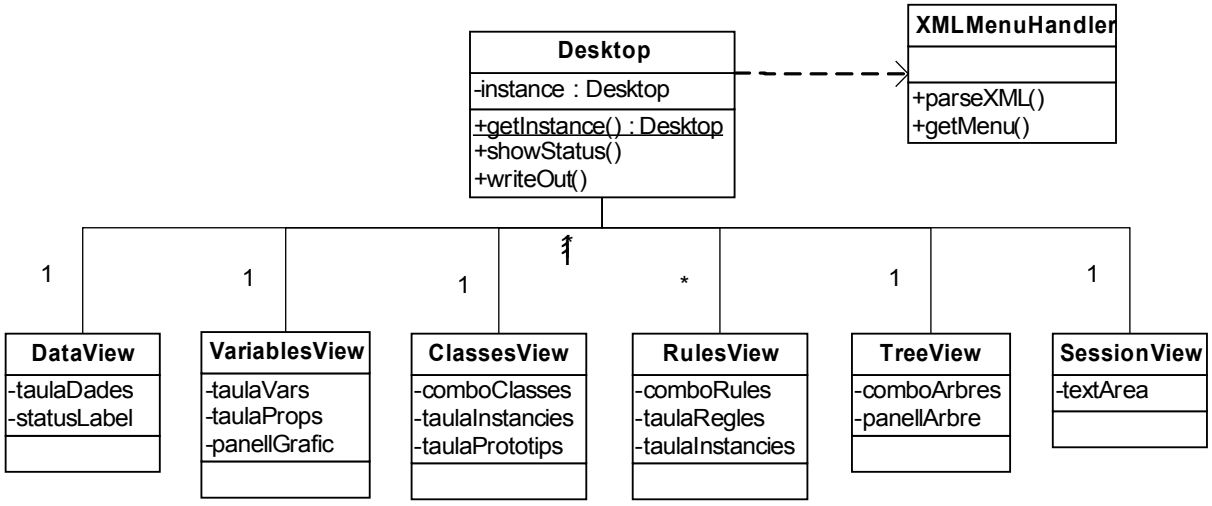


Figura 4.7: Diagrama de classes de la capa de presentació de Gesconda

Existeix una única instància de Desktop, a la que s'hi pot accedir per *Singleton* quan sigui necessari. Aquesta instància conté a la part esquerra una vista de variables, i la resta a la part dreta organitzades en pestanyes. Tots els components vista utilitzen el patró Observador per esbrinar quan es produeixen canvis sobre el model, i reaccionar a tal efecte per a reflexar aquests canvis a l'usuari.

Existeixen moltes altres classes en aquesta capa que serveixen d'ajuda a la presentació visual del model. Models, Renderers, Adapters, etc. Aquestes classes són implementacions específiques de Java swing, i no és objectiu d'aquest document explicar els detalls d'implementació de la tecnologia d'aplicacions

d'escriptori de Java. Per a més informació més detallada es pot consultar el *javadoc* inclòs en el CD de distribució de l'aplicació.

El menú de l'aplicació es construeix dinàmicament a partir d'un fitxer XML. Això permet un altíssim grau de personalització i evita al programador enfrontar-se amb centenars de classes enllaçades o bé utilitzar editors visuals de components swing que sovint embruten el codi. Des de la classe Desktop, al inicialitzar-se es carrega el fitxer XML, es *parseja* i automàticament es construeix un objecte menú amb tots els submenús, etiquetes i accions correctament enllaçades.

4.2.3. Controlador

La capa de controlador del model MVC està representada a Gesconda com un conjunt d'accions que realitzen operacions de mostrar vista (habitualment per demanar algun paràmetre), executar codi de la regla de negoci (executar l'algorisme que implementa l'acció) i notificar a la vista si s'escau els resultats obtinguts.

Les accions de Gesconda estan enllaçades als elements del menú gràcies al fitxer XML que hem comentat a l'apartat anterior. Són moltes i a més, la plataforma és fàcilment extensible, o sigui mostrar un diagrama de classes amb totes elles no aporta informació interessant. El que sí mostrarem són les classes del segon nivell del controlador, encarregades de permetre a un algorisme executar-se en un *thread* independent sense bloquejar l'aplicació:

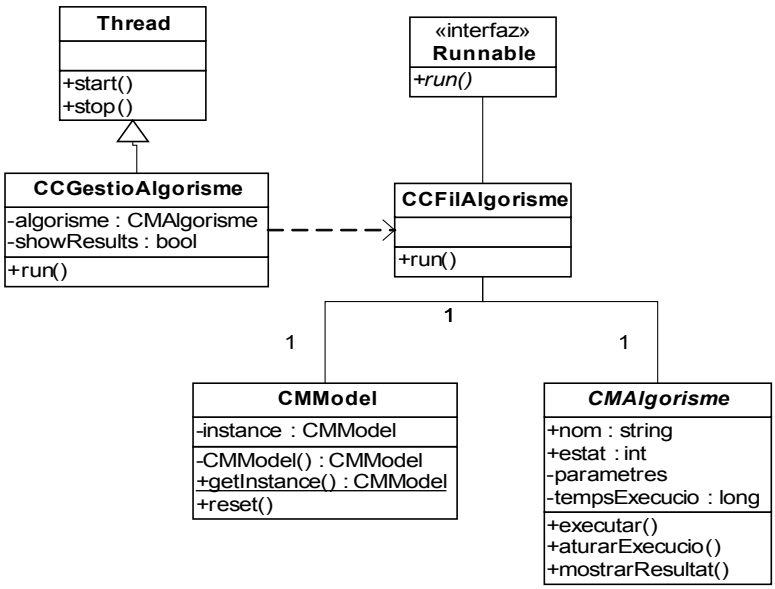


Figura 4.8: Classes de suport a l'execució d'algorismes

4.3. Disseny lògic

En aquest apartat es mostren els diagrames de casos d'ús que mostren les accions disponibles a l'usuari en l'aplicació Gesconda II. Dividirem els casos d'ús seguin la mateixa estructura en la que s'han organitzat els menús de l'aplicació, seguint la mateixa lògica emprada, en relació amb el procés d'extracció de coneixement de la mineria de dades que hem definit al primer capítol d'aquesta memòria.

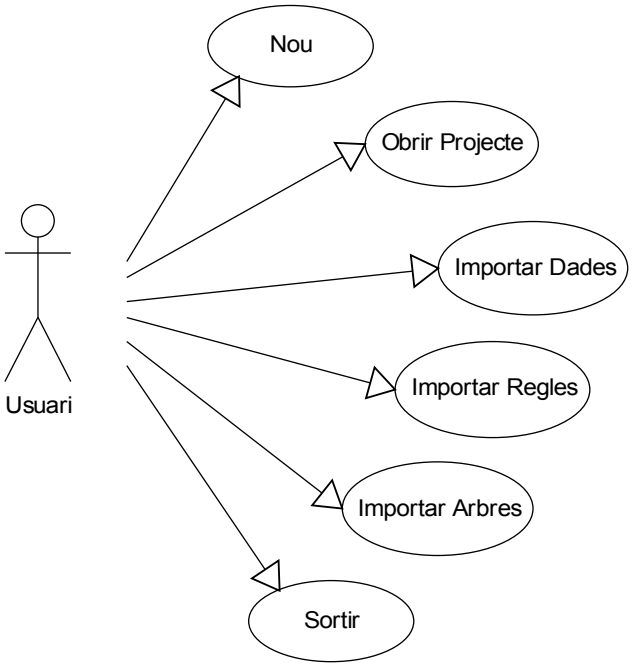


Figura 4.9: Diagrama de casos d'ús del menú arxiu

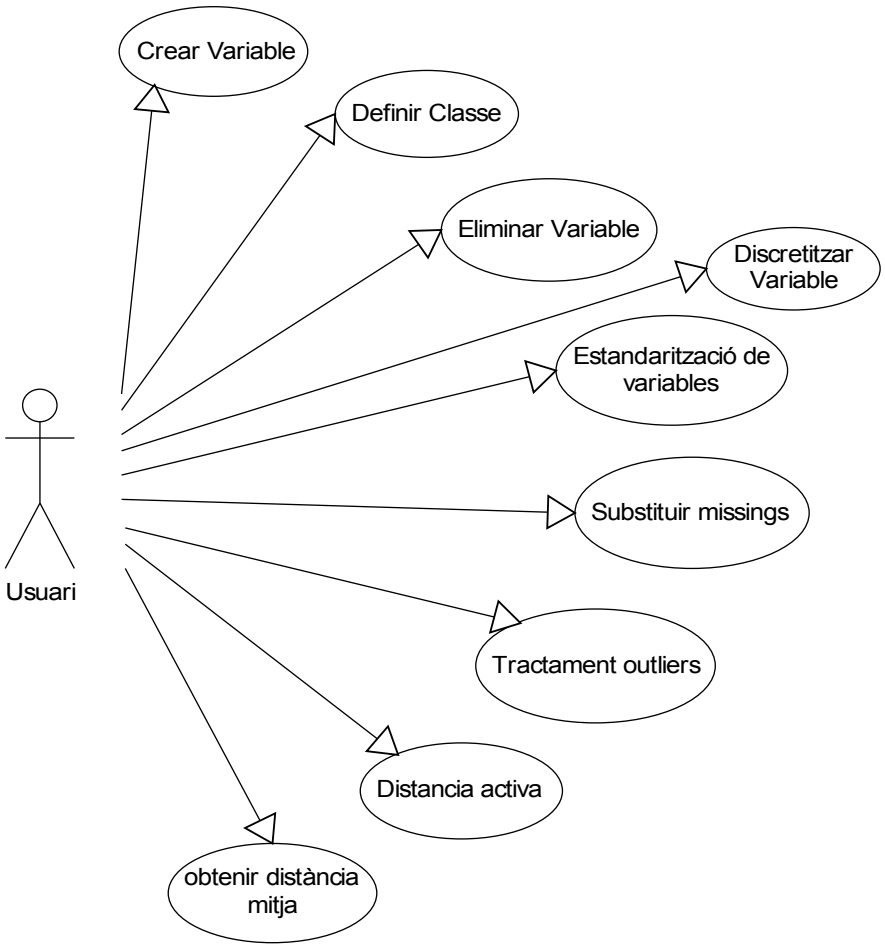


Figura 4.10: Diagrama de casos d'ús del menú Gestió de Dades

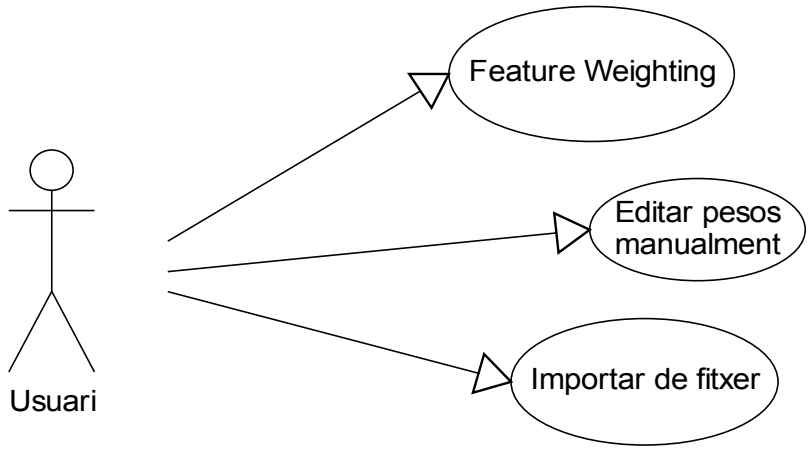


Figura 4.11: Diagrama de casos d'ús del menu Rellevància d'Atributs

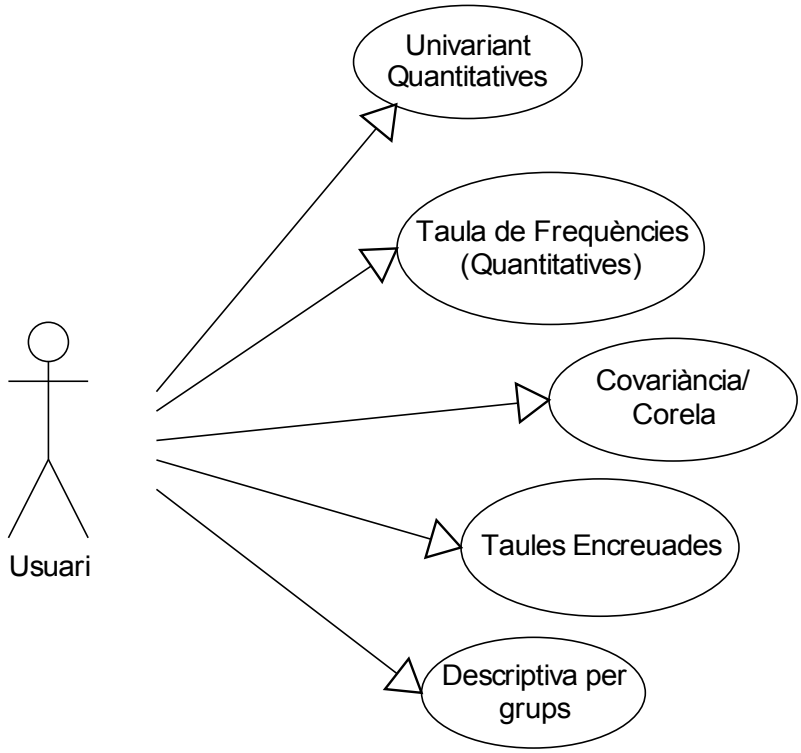


Figura 4.12: Diagrama de casos d'ús del menú d'Estadística Descriptiva

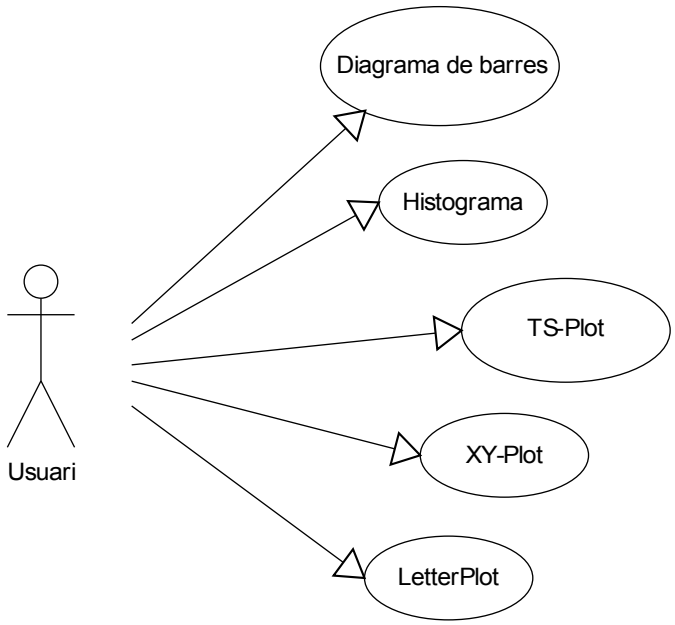


Figura 4.13: Diagrama de casos d'ús del menu Gràfics

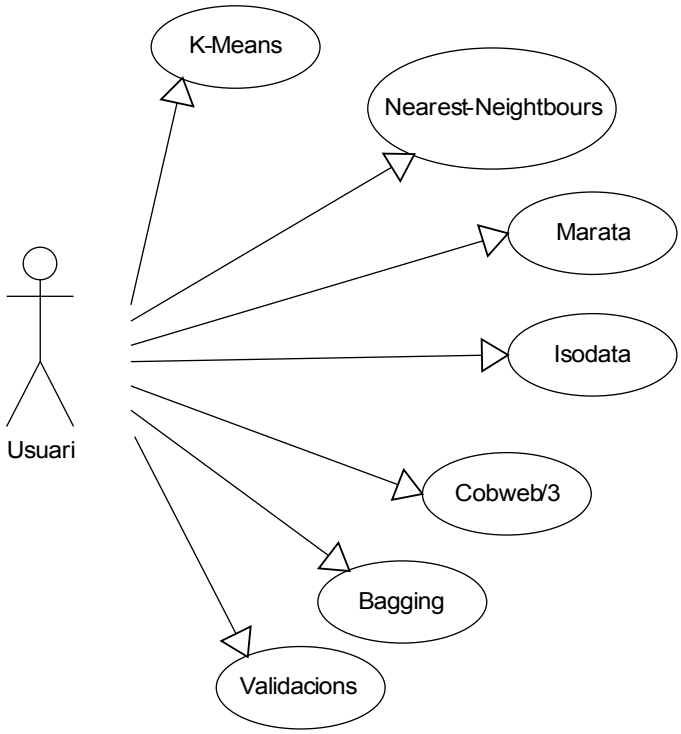


Figura 4.14: Diagrama de casos d'ús del menú Tècniques de Classificació

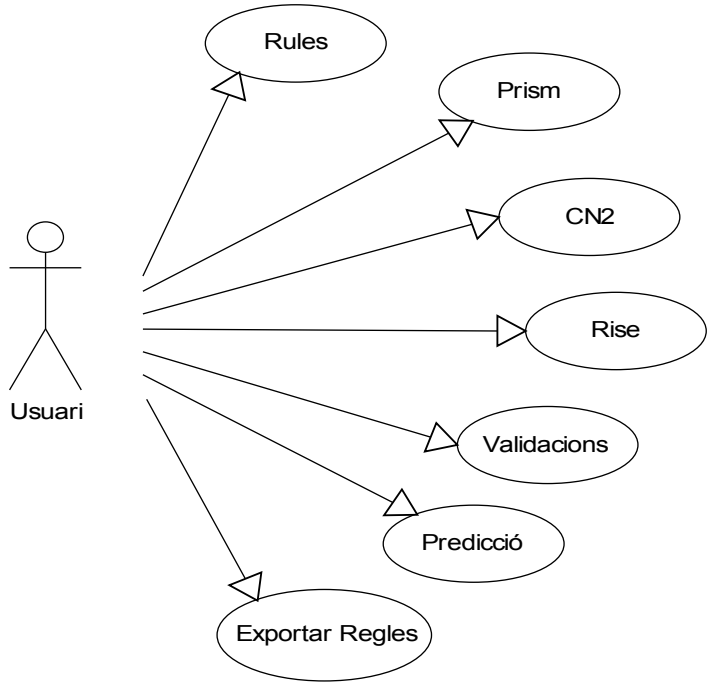


Figura 4.15: Diagrama de casos d'ús del menú Inducció de Regles

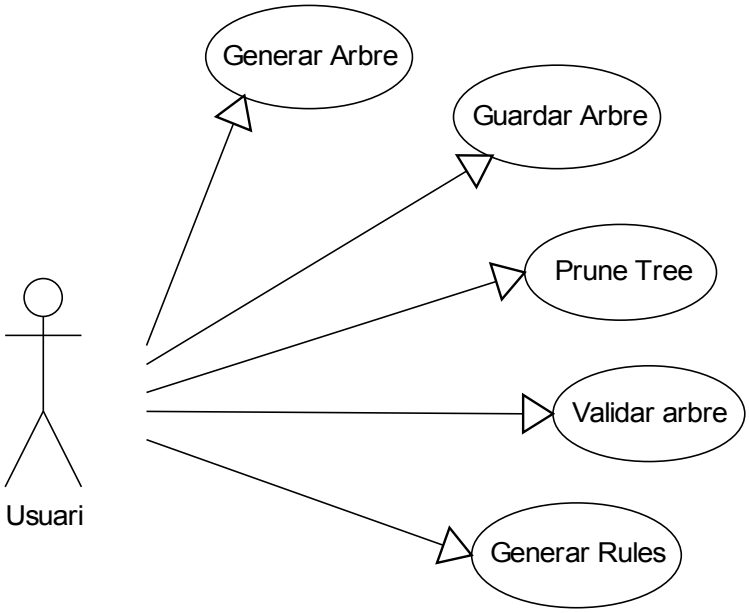


Figura 4.16: Diagrama de casos d'ús del menú Àrbres de Decisió

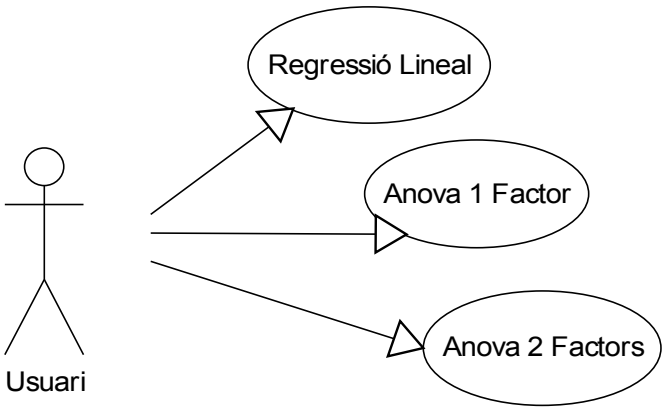


Figura 4.17: Diagrama de casos d'ús del menú Estadística Predictiva

4.4. Disseny extern

Tal i com s'ha vist abans en el disseny de les classes de la capa presentació, Gesconda II obre d'entrada una finestra principal des d'on s'executen totes les accions que posa a disposició de l'usuari. Les accions estan situades al menú superior tot i que, per facilitar el treball de l'usuari, també ofereix dreceres a algunes d'aquestes operacions directament sobre les vistes de resultats. A través de barres d'eines i menús contextuals.

4.4.1. Pantalla principal

La finestra principal de Gesconda II té aquest aspecte:

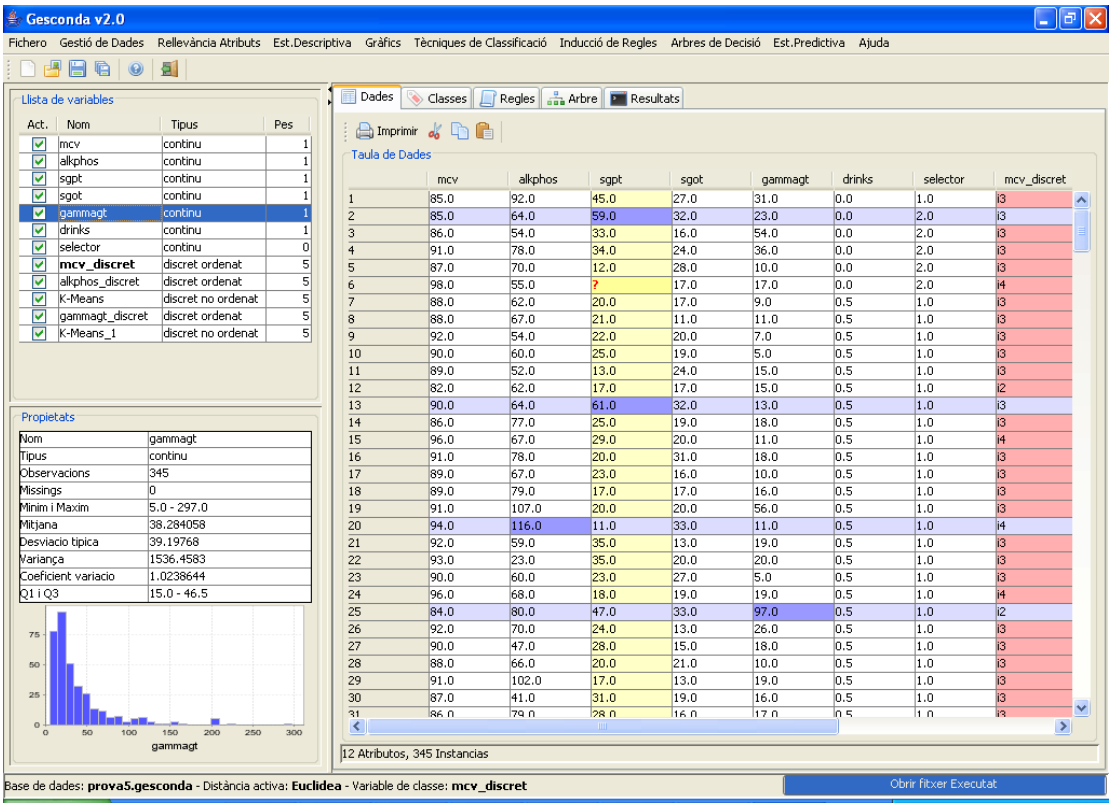
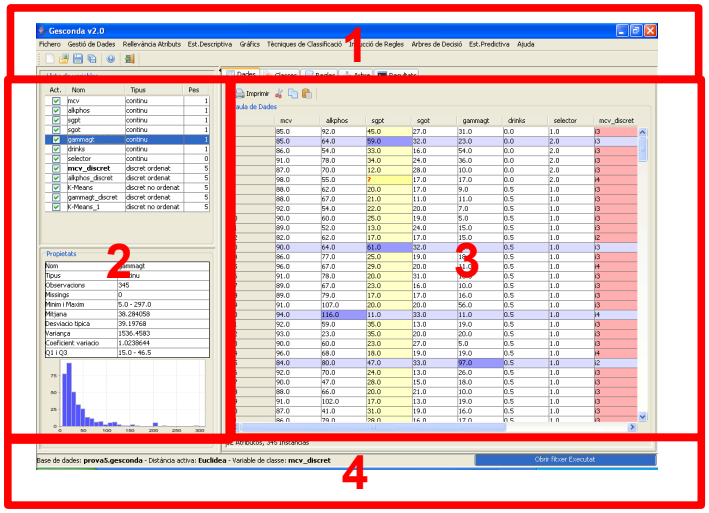


Figura 4.18: Pantalla principal de Gesconda II

A continuació es mostren els components d'aquesta pàgina principal i les seves característiques:



1. Barra de menús i barra d'eines, amb les accions que ofereix Gesconda a l'usuari. Tan el menú com la barra d'eines són configurables mitjançant un fitxer XML, fet que permet adaptar la disposició i funcionalitat de l'aplicació sense haver-ne de modificar el codi font.

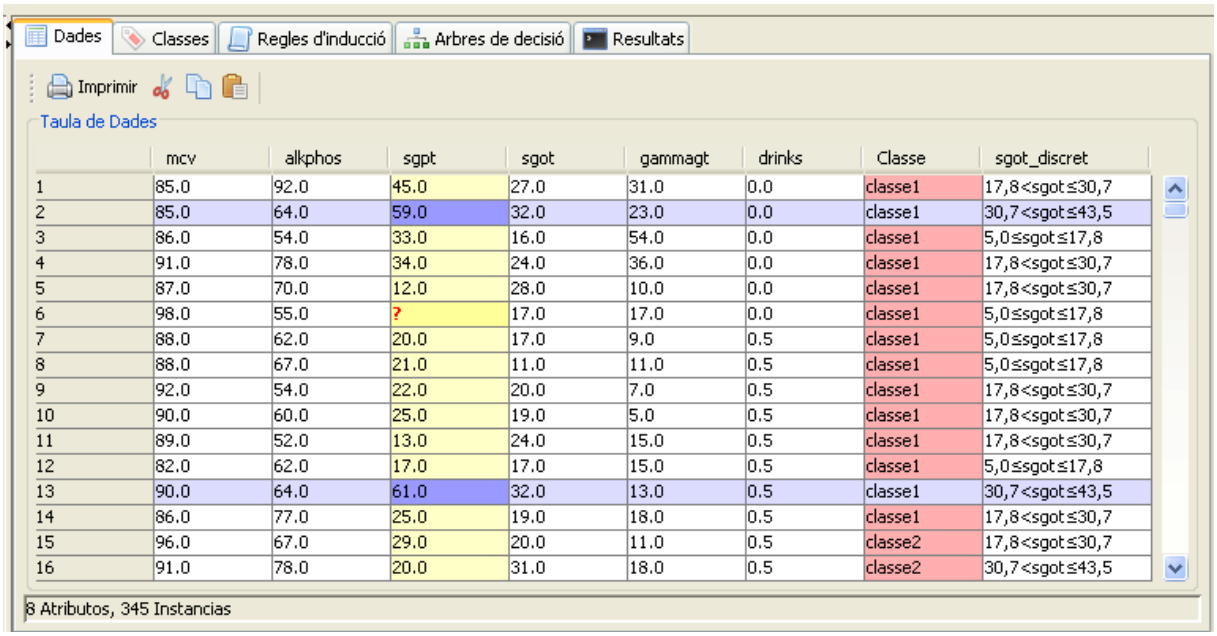
2. Vista de variables contingudes en la base de dades activa. La part superior és una llista amb les variables, des d'on es poden activar o desactivar, canviar el nom o el pes o realitzar accions sobre elles amb el menú contextual del ratolí. La part inferior és una vista de propietats de la variable seleccionada a la llista superior a més d'un gràfic representatiu dels valors de la variable.

3. Àrea principal on es mostren diferents vistes organitzades en pestanyes. Les vistes que es mostren en l'àrea principal permeten veure les dades, les classes i els seus prototips, les regles generades per els algorismes d'inducció, els arbres de decisió generats i una vista de resultats en mode text, per els algorismes que no ofereixen una representació visual. En el proper apartat es descriuen les característiques d'aquestes vistes.

4. Barra d'estat on s'indica la base de dades amb la que s'està treballant, la funció de distància activa i la variable de classe. A la part dreta hi ha una barra de progrés que informa de l'estat d'execució dels algorismes.

4.4.2. Vista de dades

La vista de dades ofereix en tot moment els valors de les instàncies i atributs actius a la base de dades amb la que s'està treballant:



	mcv	alkphos	sgpt	sgot	gammagt	drinks	Classe	sgot_discret
1	85.0	92.0	45.0	27.0	31.0	0.0	classe1	17,8<sgot≤30,7
2	85.0	64.0	59.0	32.0	23.0	0.0	classe1	30,7<sgot≤43,5
3	86.0	54.0	33.0	16.0	54.0	0.0	classe1	5,0≤sgot≤17,8
4	91.0	78.0	34.0	24.0	36.0	0.0	classe1	17,8<sgot≤30,7
5	87.0	70.0	12.0	28.0	10.0	0.0	classe1	17,8<sgot≤30,7
6	98.0	55.0	?	17.0	17.0	0.0	classe1	5,0≤sgot≤17,8
7	88.0	62.0	20.0	17.0	9.0	0.5	classe1	5,0≤sgot≤17,8
8	88.0	67.0	21.0	11.0	11.0	0.5	classe1	5,0≤sgot≤17,8
9	92.0	54.0	22.0	20.0	7.0	0.5	classe1	17,8<sgot≤30,7
10	90.0	60.0	25.0	19.0	5.0	0.5	classe1	17,8<sgot≤30,7
11	89.0	52.0	13.0	24.0	15.0	0.5	classe1	17,8<sgot≤30,7
12	82.0	62.0	17.0	17.0	15.0	0.5	classe1	5,0≤sgot≤17,8
13	90.0	64.0	61.0	32.0	13.0	0.5	classe1	30,7<sgot≤43,5
14	86.0	77.0	25.0	19.0	18.0	0.5	classe1	17,8<sgot≤30,7
15	96.0	67.0	29.0	20.0	11.0	0.5	classe2	17,8<sgot≤30,7
16	91.0	78.0	20.0	31.0	18.0	0.5	classe2	30,7<sgot≤43,5

Figura 4.19: Vista de dades

Ofereix en una barra d'eines a dalt un botó per a imprimir la matriu de dades, i les opcions típiques per a l'edició (tallar, copiar i enganxar).

La taula mostra en diferents colors les característiques dels valors mostrats:

- En color vermell es destaca l'atribut classe de la base de dades.
- Els atributs que contenen valors *missing* es mostren en color groc clar, la cel·la amb el valor missings és d'un groc més fosc i amb el text en vermell.
- Les instàncies que contenen valors *outlier* es marquen en color blau clar, en un color blau més fosc el valor *outlier* en concret, es diferencia *l'outlier* extrem del suau perquè el primer mostra el valor en negreta.

Es poden editar els valors directament sobre la taula, així com copiar i enganxar des d'altres aplicacions, com per exemple MS Excel. L'usuari també pot reordenar les

columnes que es mostren en funció de les seves preferències.

A la part inferior de la vista s'hi mostra una barra d'estat amb informació sobre número d'atribut i número d'instàncies que conté la base de dades.

4.4.3. Vista de classes

La vista de classes permet veure els prototips que representen els valors d'una variable discreta, i les instàncies que s'associen a cada prototip.

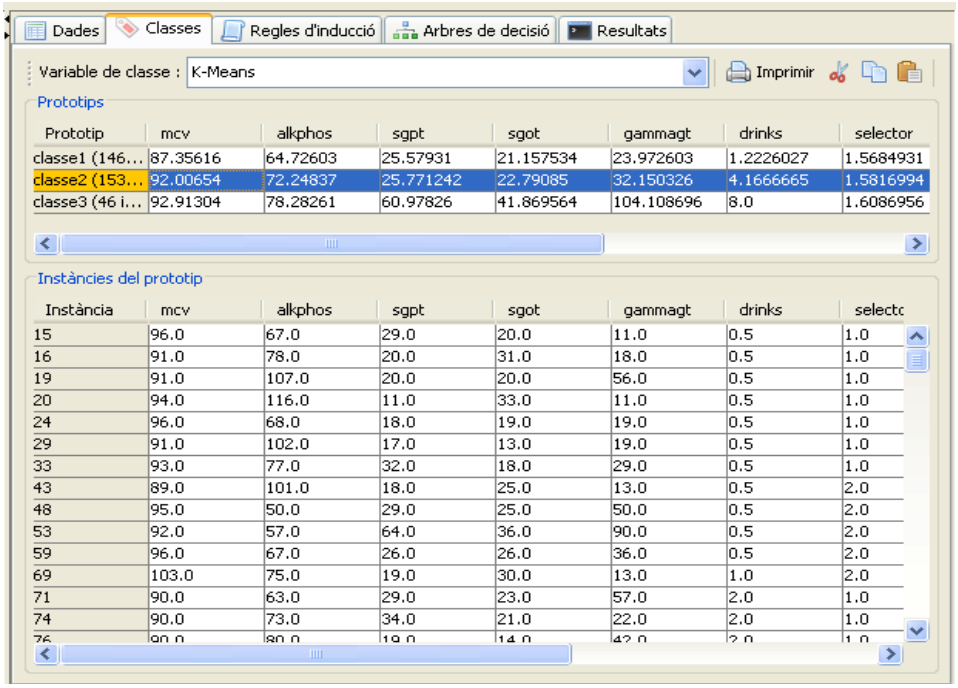


Figura 4.20: Vista de classes

4.4.4. Vista de regles

La vista de regles permet veure els resultats dels algorismes d'inducció. La *combo* que apareix a la part superior incorpora les execucions d'aquests algorismes i al seleccionar un d'ells es mostren a la taula les regles generades. Si seleccionem una regla apareix a la taula inferior la llista d'instàncies cobertes per aquesta regla.

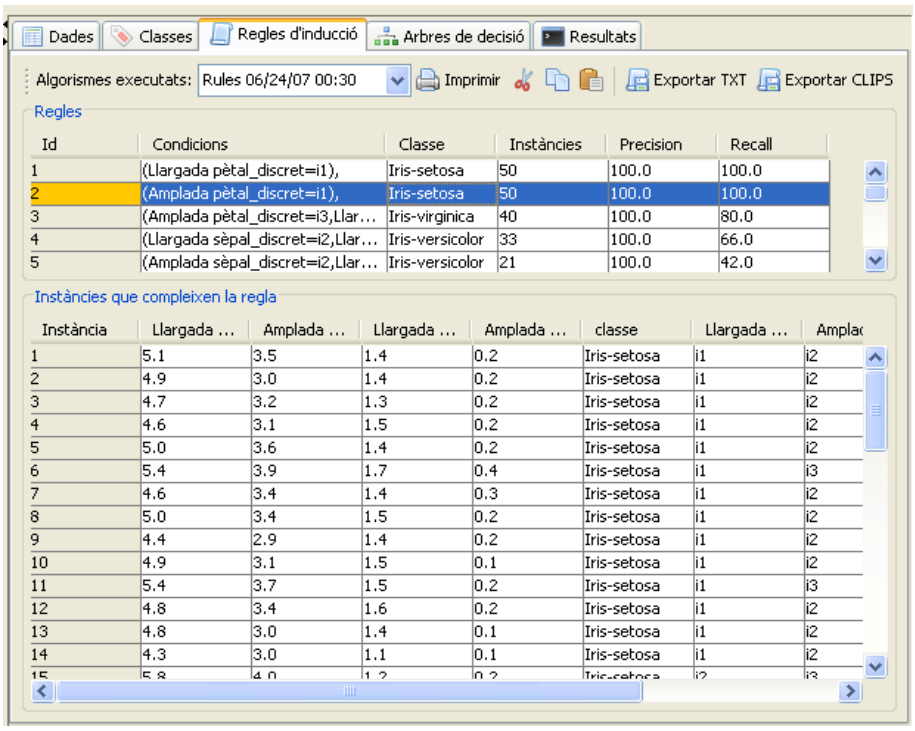


Figura 4.21: Vista de regles

4.4.5. Vista d'arbres

La vista d'arbres de decisió mostra una representació gràfica dels arbres generats per l'aplicació. Permet, des de la barra d'eines, generar regles a partir de l'arbre, o bé executar accions de post-prunning sobre l'arbre seleccionat.

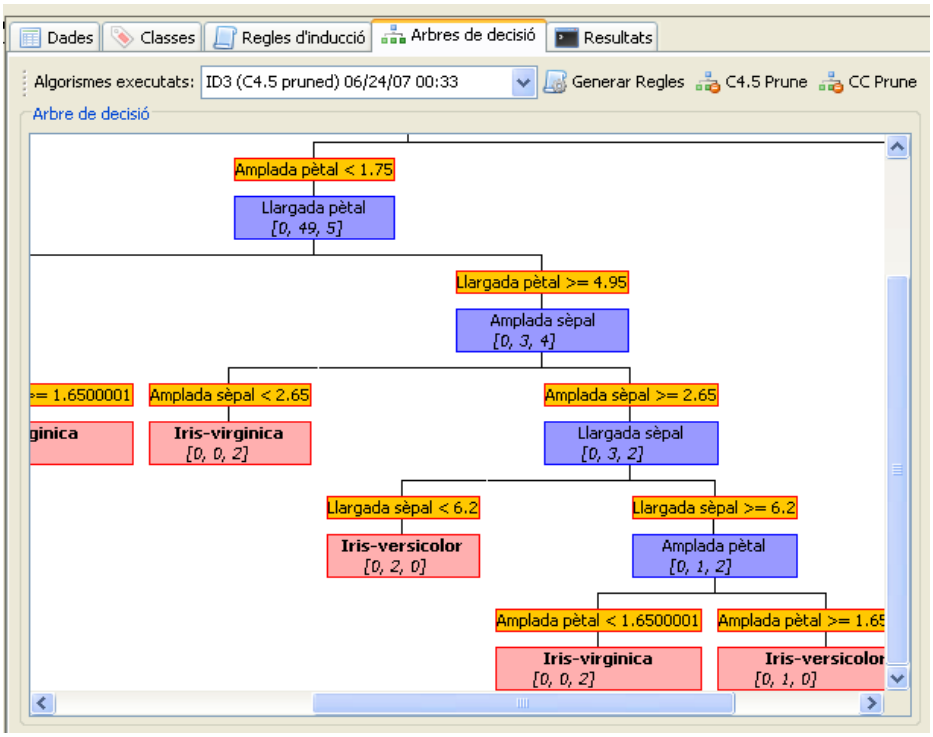


Figura 4.22: Vista d'arbres de decisió

4.4.6. Vista de resultats

Per últim, la vista de resultats, que mostra en mode text resultats obtinguts per algorismes que no ofereixen representació visual. Poden ser impresos, exportats a fitxer o copiats al porta-papers del sistema per enganxar-los en una altra aplicació.

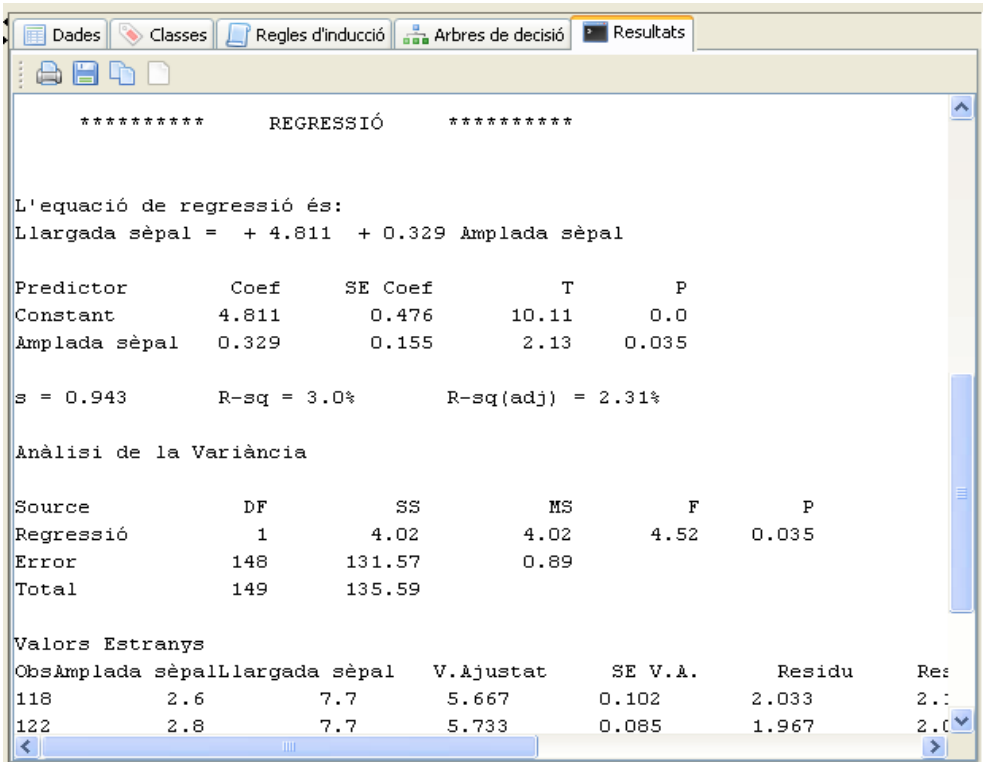


Figura 4.23: Vista de resultats

5. Validació de l'aplicació

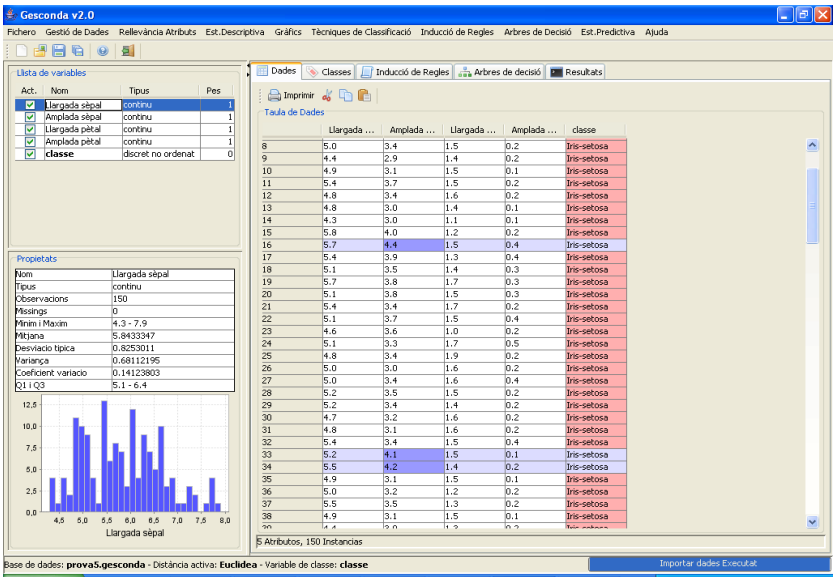
Durant el desenvolupament del projecte, i seguint la metodologia descrita en l'apartat 2.6, s'han realitzat proves unitàries de cadascuna de les funcionalitats a implementar. Tot i això, un cop acabat el desenvolupament de totes les àrees funcionals requerides, es fa necessari executar proves d'integració sobre el conjunt de l'aplicació. En aquest apartat es mostra el resultat d'una d'elles on es realitza un procés complert de mineria de dades executant varis algorismes de diferents tipus.

Per a realitzar el cas de prova hem fer servir la base de dades **iris**, de l'UCI *Machine Learning Database Repository*. Aquesta base de dades conté 150 de diferents tipus de plantes. Els seus atribut són:

- Llargada sèpal: És un atribut continu amb valors reals corresponents a la longitud del sèpal en cm.
- Amplada sèpal: És un atribut continu amb valors reals corresponents a l'amplada del sèpal en cm.
- Llargada pètal: És un atribut continu amb valors reals corresponents a la longitud del pètal de la planta en cm.
- Amplada pètal: És un atribut continu amb valors reals corresponents a l'amplada del pètal de la planta en cm.
- classe: És un atribut discret i pren tres valors possibles que es corresponen amb al tipus de planta al que pertany la instància. Els valors que pren aquest atribut són: Iris Setosa, Iris Versicolor i Iris Virgínica.

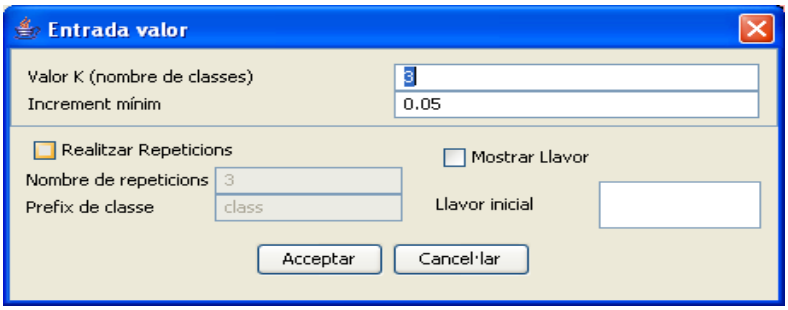
5.1. Validació de clustering

Començarem a realitzar les proves executant algorismes de clustering. En aquest cas no té massa sentit aplicar algorismes a una base de dades d'aquest estil, perquè ja disposem d'un atribut classe conegut. No obstant, el desactivarem i provarem de generar-ne un de nou que hauria de ser bastant similar:

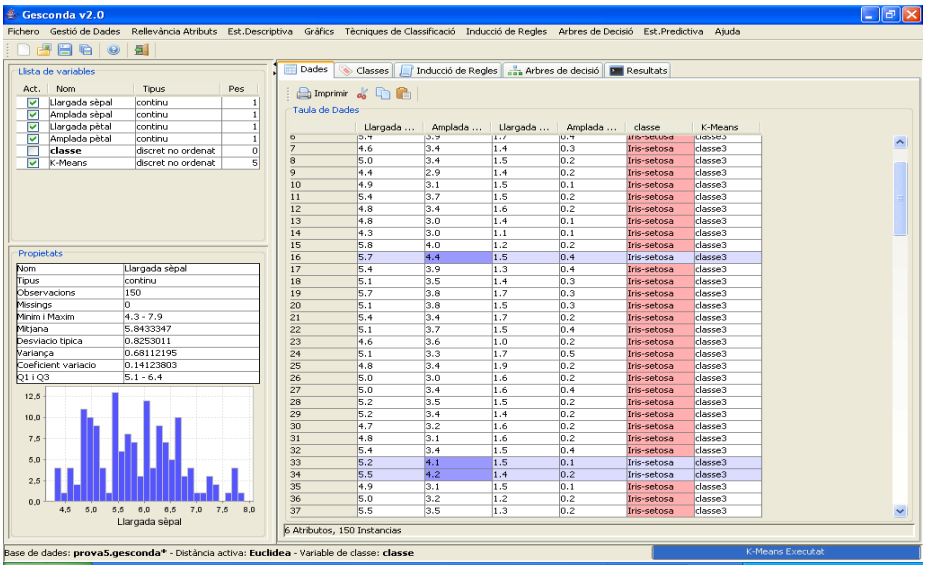


A simple vista observem que tenim alguns valors *outliers* (en blau) i que no tenim cap valor *missing*. Ignorarem en aquest cas els primers al tractar-se d'*outliers* suaus. Procedim a desactivar la variable classe i executar, per exemple, l'algorisme de clustering K-Means:

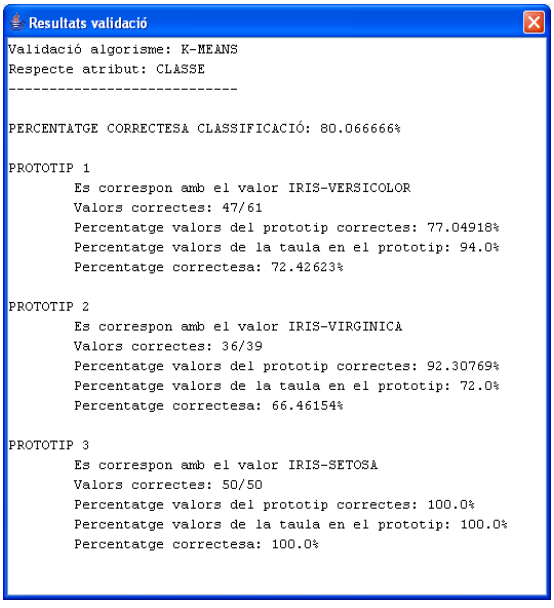
Informarem que volem trobar 3 classes (fem trampa perquè ja sabem que n'hi ha tres) llencem l'execució de l'algorisme.



Observem que ens ha aparegut un nou atribut corresponent a la classificació que ha trobat K-Means:

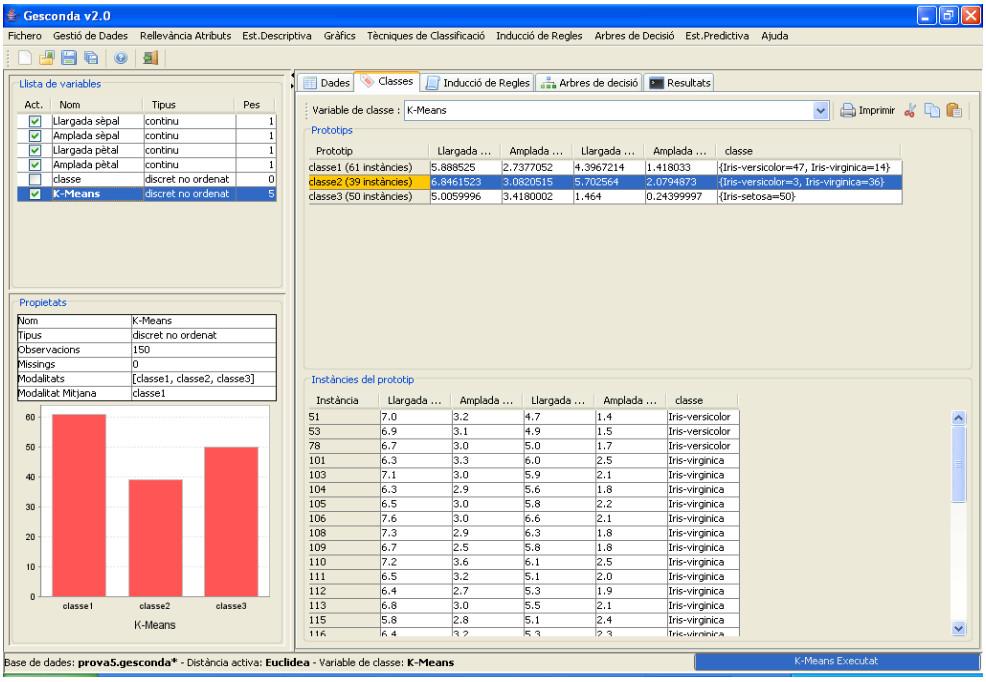


Si procedim a validar el resultat obtingut ens apareix, de la mateixa manera que ho feia l'anterior mòdul de clustering, aquesta finestra:



El percentatge d'encert és bastant elevat, ja que una de les classes (iris-setosa) és linealment separable.

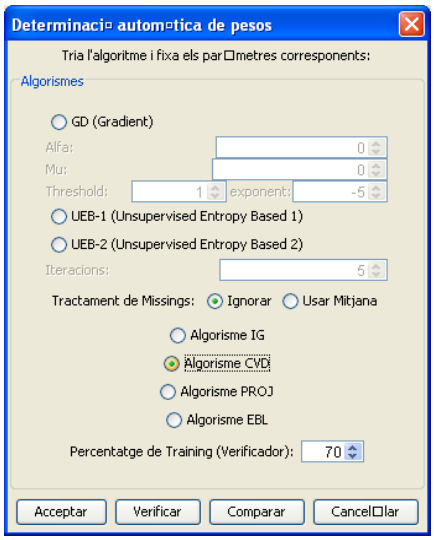
Ara, en el visor de classes podem seleccionar el nou atribut generat i visualitzar com són els prototips generats i quines instàncies queden cobertes:

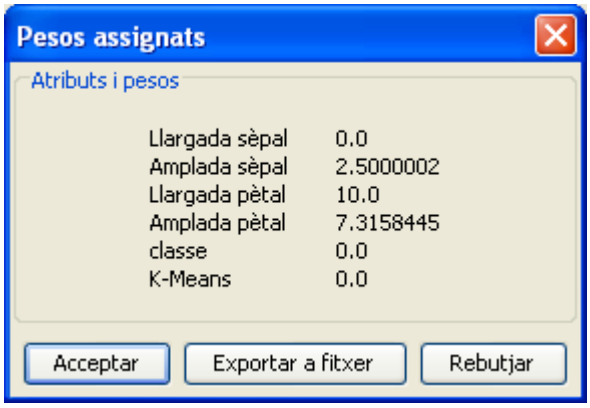


A la part esquerra (el visor de variables) hem seleccionat també aquest atribut i ens apareix el diagrama de barres corresponent als valors i freqüències de l'atribut generat.

5.2. Validació de Feature Weighting

Continuant amb el cas de prova anterior, l'atribut nou generat ha estat creat amb un pes valor 5 (valor per defecte), anem ara a executar un algorisme de Feature Weighting per esbrinar els pesos de la resta d'atributs:

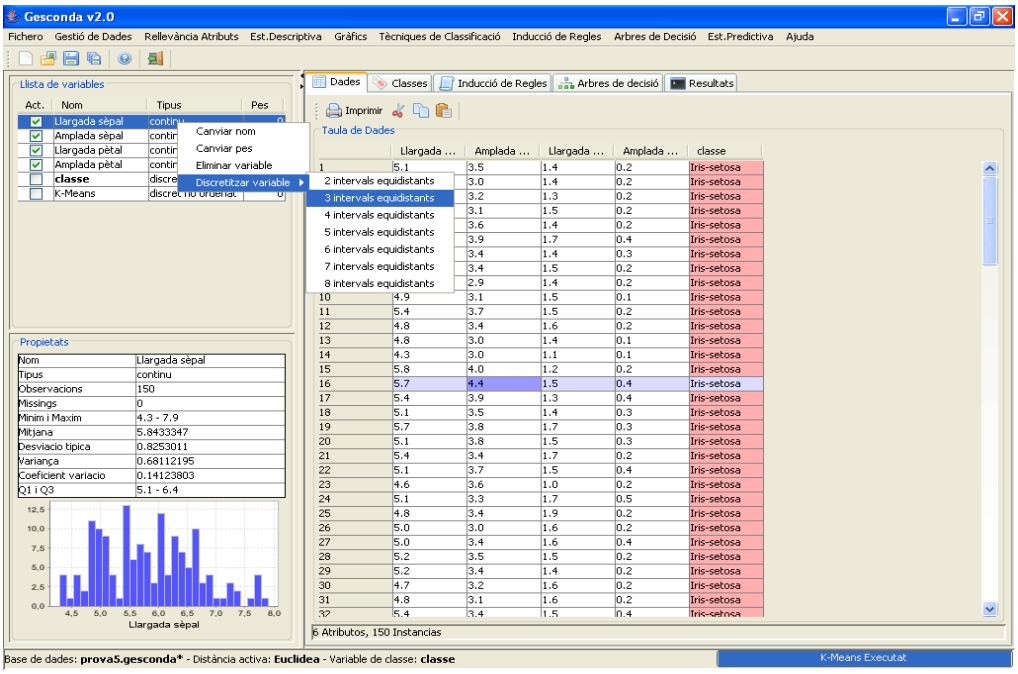




En el resultat obtingut, trobem l'explicació al fet que abans K-Means hagi trobat les classes amb tanta facilitat. Segons l'algorisme de Feature Weighting executat, els atributs de l'amplada, i sobretot la llargada del pètal tenen una gran rellevància en la base de dades.

5.3. Validació de Regles

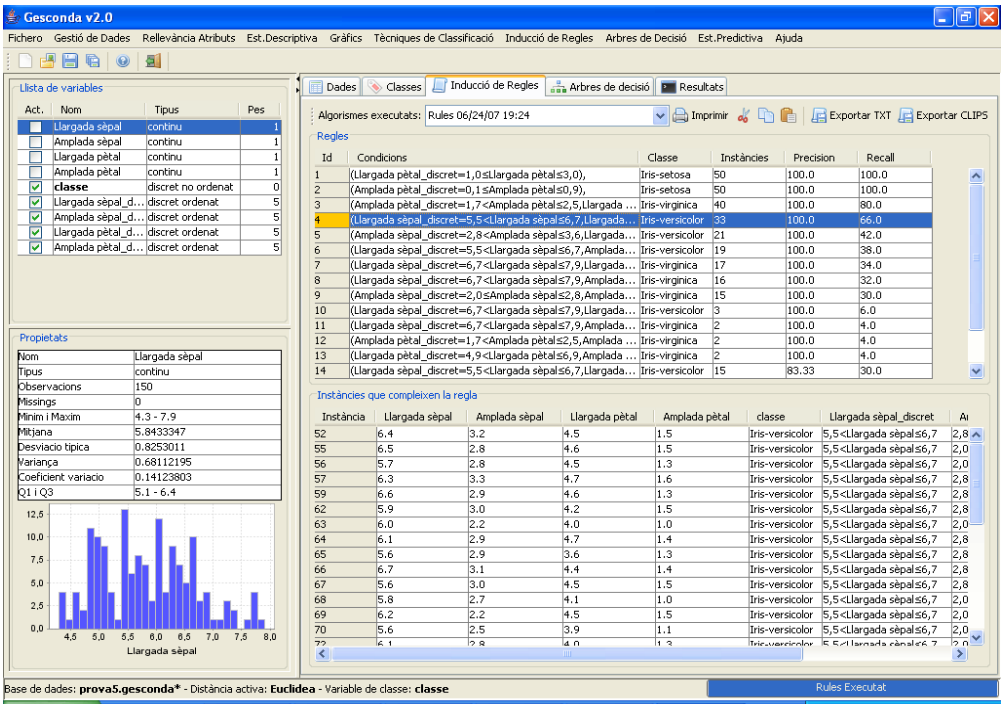
Passem ara a validar els algorismes de regles, però abans haurem de discretitzar els atributs continus:



El primer atribut l'hem discretitzat en 3 intervals equidistants. En aquesta ocasió ho hem fet utilitzant el menú contextual del ratolí sobre la vista de variables, però també ho haguéssim pogut executar des del menú principal, des d'on disposem de moltes més opcions:

La discretització personalitzada ens permet acurar quins són els valors de tall de la discretització, a més de poder variar els valors que prendrà la nova variable.

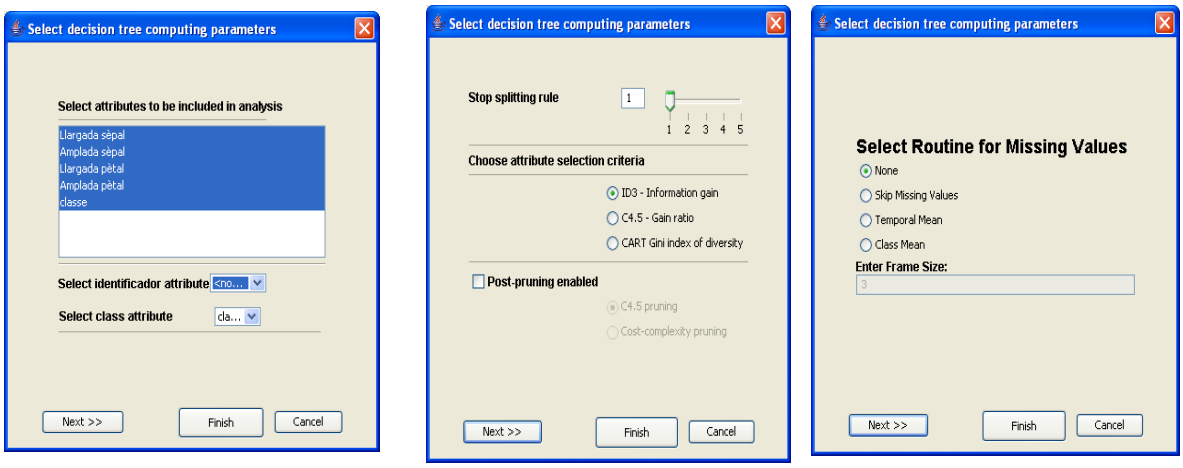
Procedim ara a l'execució de l'algorisme Rules d'inducció de regles. No ens demana cap paràmetre i genera directament aquest resultat:



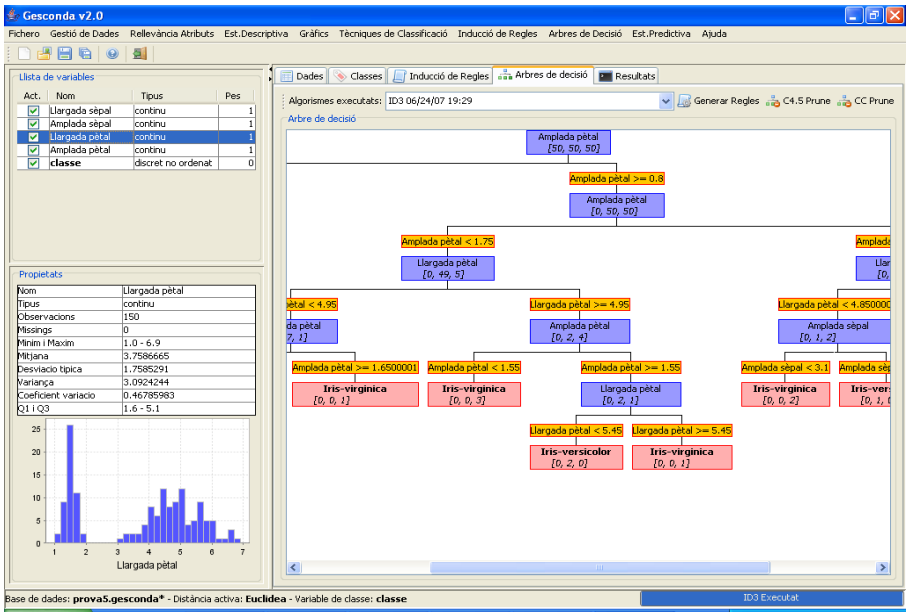
En la vista de regles, ens apareix l'execució de Rules amb la data i hora i, quan la seleccionem, carrega les regles generades. Per cada regla, veiem l'antecedent (les condicions), el conseqüent (la classe induïda), el nombre d'instàncies cobertes i els percentatges de *recall* i *accuracy*. Un cop seleccionem alguna de les regles generades, ens apareix a la part inferior la llista d'instàncies cobertes per la regla.

5.4. Validació d'Arbres de Decisió

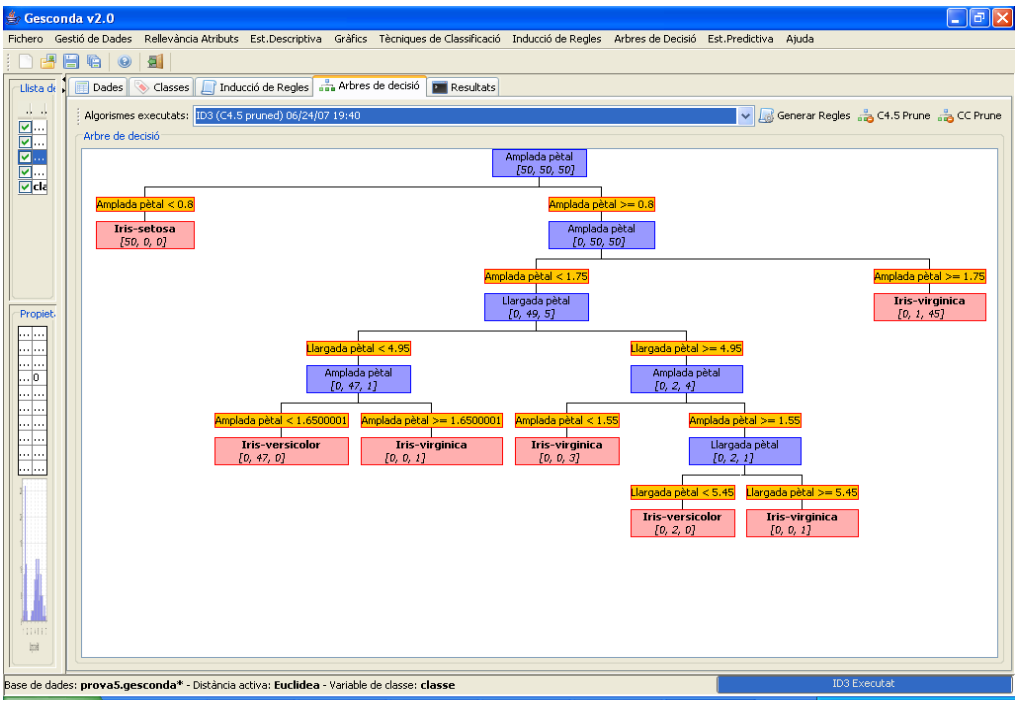
Executem ara un algorisme d'arbres de decisió. Al seleccionar la opció de generar arbre al menú principal, ens apareix un diàleg tipus *wizard* amb diferents passos:



Seleccionarem en aquesta ocasió l'algorisme ID3, sense marcar cap opció de *Post-prunning*, ja que ara l'aplicació ens permet executar-la a posteriori, sobre l'arbre generat. L'arbre es mostra a la vista d'arbres de decisió d'aquesta manera:



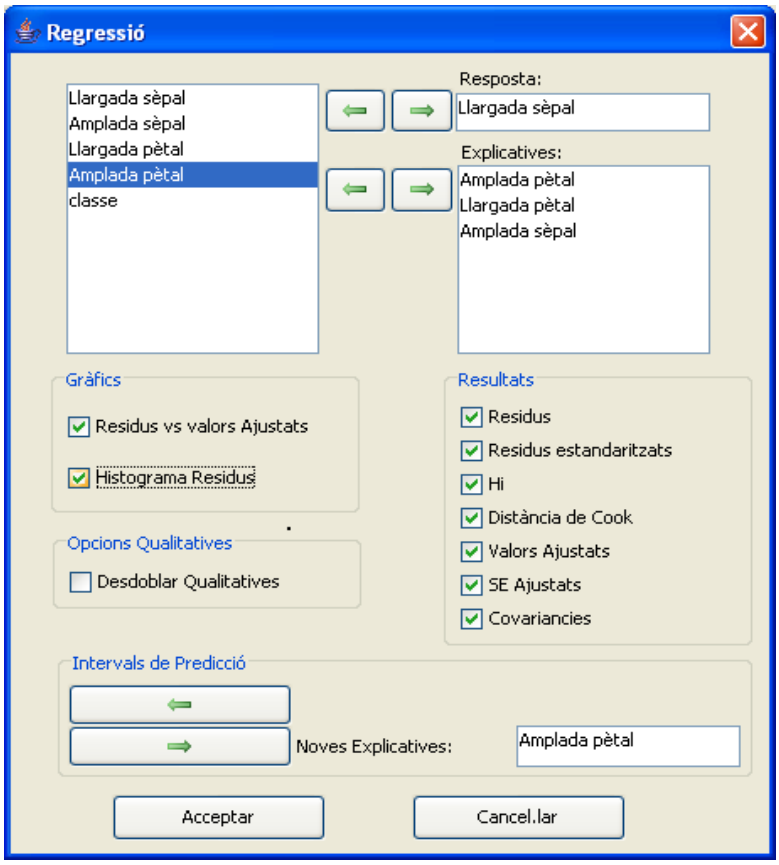
Un cop tenim l'arbre generat, podem aplicar-li algun dels algorismes de *post-prunning*, de manera que se'ns generarà un nou arbre amb la poda de nodes aplicada. Veiem en aquest cas com queda l'arbre obtingut en el pas anterior després d'executar una poda:



Hem desplaçat el visor de variables a l'esquerra per a què l'arbre podat aparegui sencer en pantalla. Es pot apreciar com la part de la dreta ha estat reduïda a un sol node, quan abans n'eren cinc.

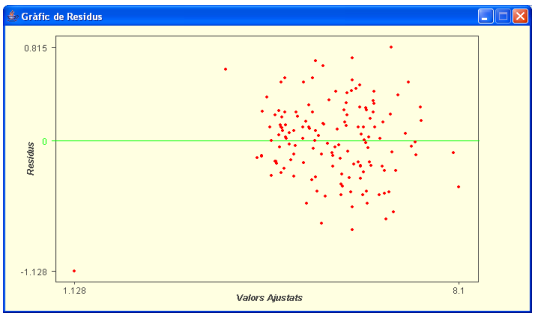
5.5. Validació de l'Estadística predictiva

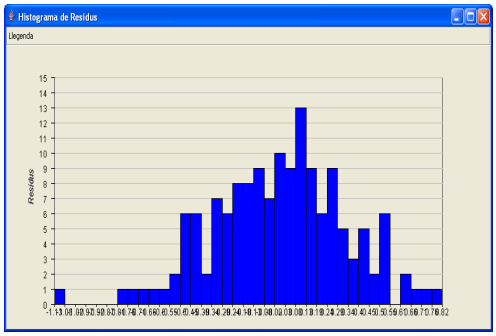
Aquest nou menú, correspon a funcionalitat anteriorment programada a Gesp. Anem a executar una regressió lineal tal i com es mostra en el següent formulari:



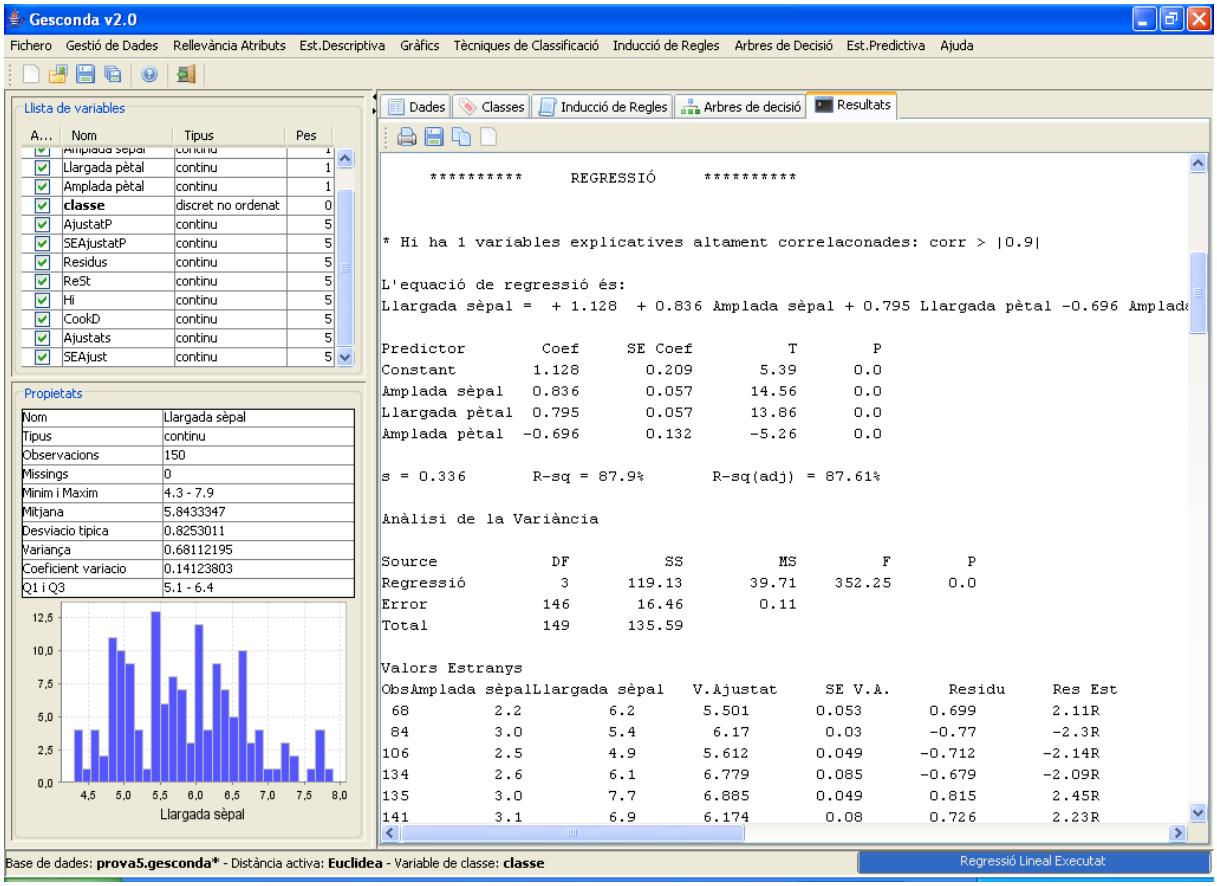
Li demanarem que ens generi tots els resultats possibles, a més de mostrar-nos gràfics de residus i ajustats.

Ens apareixen les finestres amb els gràfics demanats:





I a més, genera els resultats a la vista en mode text:



Podem observar també a la part esquerra que ha generat noves variables corresponents a tots els resultats que li hem demanat al principi.

6. Manual d'Usuari

6.1. Instal·lació

L'aplicació no requereix de cap instal·lació especial per a ser executada. Es lliura en una distribució compactada de manera que es pot executar fent doble clic sobre l'arxiu proporcionat. Per a què funcioni cal tenir correctament instal·lat una versió 5 o superior del Runtime de Java.

Un com hem fet doble clic sobre l'arxiu que conté l'aplicació Gesconda, l'aplicació s'obre automàticament i a partir d'aquest moment ja podem començar a treballar important arxius de dades per tal d'extreure'n informació.

6.2. Pantalla principal

Gesconda II s'inicia obrint aquesta pantalla, que permet executar tota la operativa que incorpora el producte. Accedirem a la funcionalitat desitjada utilitzant el menú principal situat a la part superior de la finestra. Per a major comoditat, disposa d'una barra d'eines també a la part superior des d'on podrà executar de forma ràpida les operacions més habituals.

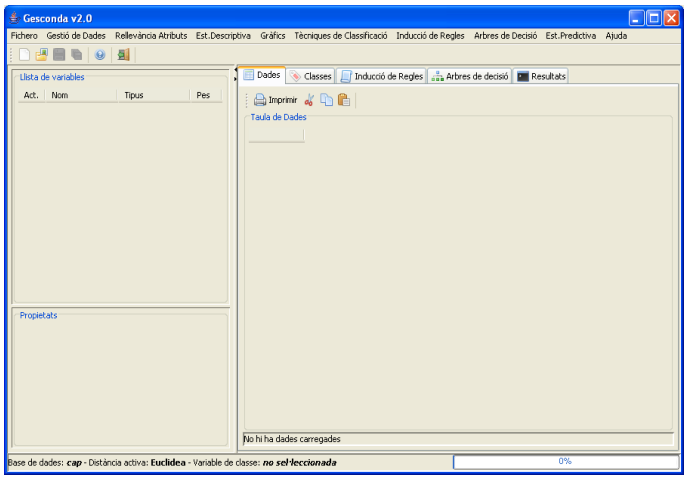


Figura 6.1: Pantalla principal de Gesconda II

A la figura 6.1, pot veure's la ubicació del menú principal a la part superior i la de la barra d'eines, situada just a sota. A la part inferior hi trobarà una barra d'estat i una àrea de notificacions, on es mostra l'estat d'execució dels algorismes en procés. A la barra d'estat s'hi mostra en tot moment la base de dades carregada, la funció de distància activa seleccionada i la variable de classe.

L'àrea principal de la pantalla està dividida en dues parts. La part esquerra (figura 6.2) conté una llista on podrà veure les **variables** que conté la base de dades carregada, i just a sota una taula on es mostren les **propietats** de la variable seleccionada a la part superior. L'àrea de propietats de la variable seleccionada també conté un **gràfic** representatiu dels valors de la variable, mostrant un histograma si es tracta d'una variable quantitativa o un diagrama de barres si és qualitativa.

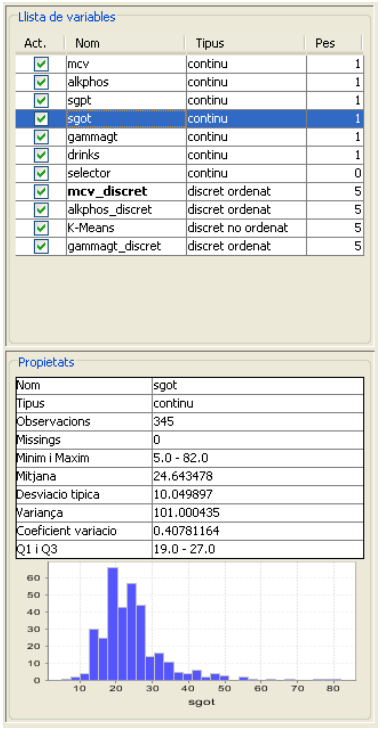
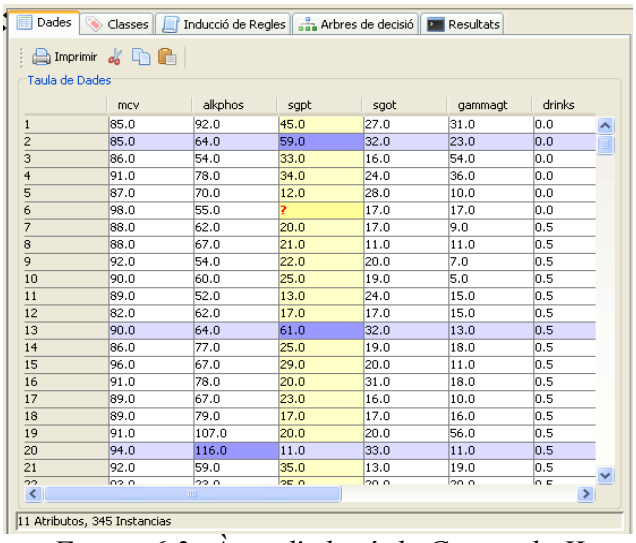


Figura 6.2: Vista de variables de l'àrea principal

La part dreta de l'àrea principal, anomenada àrea d'edició es on s'ubica la major part de la funcionalitat de l'aplicació. Està organitzada en pestanyes de manera que el seu contingut varia segons la pestanya seleccionada. Les vistes que poden mostrar-se en aquesta zona son:

- **Dades**: mostra i permet editar els valors de la graella de dades.
- **Classes**: permet seleccionar la variable de classe i visualitzar els prototips que genera, juntament amb les instàncies de cada prototip.
- **Inducció de regles**: s'hi mostra els resultats dels algorismes d'inducció i permet visualitzar les regles generades i les instàncies cobertes per cada regla.



	mcv	alkphos	sgpt	sgot	gammagt	drinks
1	85.0	92.0	45.0	27.0	31.0	0.0
2	85.0	64.0	59.0	32.0	23.0	0.0
3	86.0	54.0	33.0	16.0	54.0	0.0
4	91.0	78.0	34.0	24.0	36.0	0.0
5	87.0	70.0	12.0	28.0	10.0	0.0
6	98.0	55.0	?	17.0	17.0	0.0
7	88.0	62.0	20.0	17.0	9.0	0.5
8	88.0	67.0	21.0	11.0	11.0	0.5
9	92.0	54.0	22.0	20.0	7.0	0.5
10	90.0	60.0	25.0	19.0	5.0	0.5
11	89.0	52.0	13.0	24.0	15.0	0.5
12	82.0	62.0	17.0	17.0	15.0	0.5
13	90.0	64.0	61.0	32.0	13.0	0.5
14	86.0	77.0	25.0	19.0	18.0	0.5
15	96.0	67.0	29.0	20.0	11.0	0.5
16	91.0	78.0	20.0	31.0	18.0	0.5
17	89.0	67.0	23.0	16.0	10.0	0.5
18	89.0	79.0	17.0	17.0	16.0	0.5
19	91.0	107.0	20.0	20.0	56.0	0.5
20	94.0	116.0	11.0	33.0	11.0	0.5
21	92.0	59.0	35.0	13.0	19.0	0.5

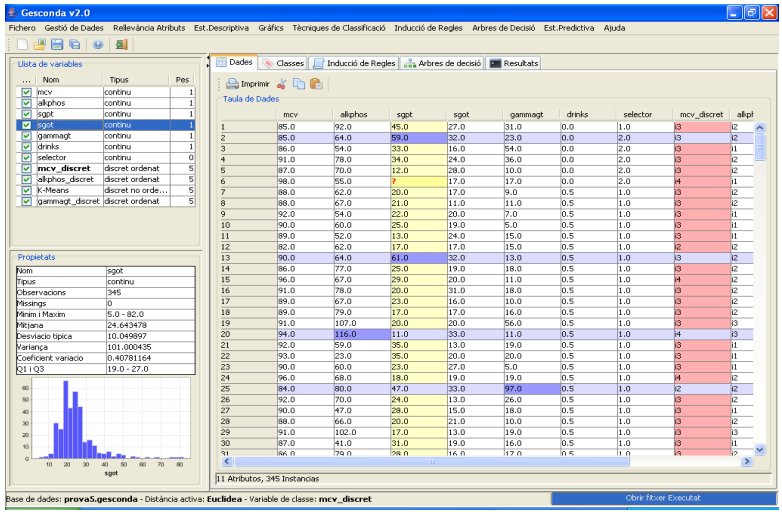
Figura 6.3: Àrea d'edició de Gesconda II

- **Arbres de decisió**: s'hi mostren els arbres que generen els algorismes d'aquest tipus. Permet generar regles a partir de l'arbre, o executar algorismes de post-prunning sobre ells.
- **Resultats**: mostra resultats dels algorismes en mode text.

6.3. Manipulant les dades

La primera acció que cal realitzar per treballar amb Gesconda II, és obrir alguna base de dades de la què en vulguem extreure informació. Per aconseguir-ho utilitzarem el menú *Arxiu, Obrir*. Ens apareixerà un quadre de diàleg on podrem seleccionar un fitxer.

Un cop el fitxer estigui carregat, es mostren les variables a la part esquerra i les dades a la part dreta, tal i com es veu en la imatge.



A la part superior de la graella de dades hi disposa d'una barra d'eines des d'on podrà imprimir les dades, a més de realitzar operacions comunes d'edició (tallar, copiar i enganxar). Pot seleccionar els valors per files, columnes, o rangs de cel·les amb el ratolí.

Les cel·les que apareixen colorejades tenen un significat especial que es detalla a continuació:

En color **blau clar** es mostren les instàncies que contenen algun valor *outlier*. El valor outlier es mostra marcat amb un color **blau més fosc**, i el tipus de lletra és normal si es tracta d'un *outlier* suau, o en negreta si es tracta d'un *outlier* extrem.

Els atributs que contenen algun valor *missing* es mostren colorejats en **color groc suau**. Podrem identificar el valor que manca perquè estarà marcat amb un color **groc més intens**.

La columna corresponent a l'atribut classe es mostra en color **vermell**.

Al menú de *Gestió de dades* hi trobarà les operacions habituals per a gestionar els valors de la base de dades:

- **Crear variable**: afegeix noves variables a la base de dades, ja sigui omplint els valors manualment, numerant-los seqüencialment o bé prenent-lo de diverses variables estadístiques: Bernoulli, Binomial, Discreta, Uniforme, Normal, Poisson i Exponencial.

- **Definir classe**: permet seleccionar la variable de classe. Aquesta operació també es pot realitzar amb el menú contextual del ratolí sobre la llista de variables de la base de dades.

- **Eliminar variable**: permet eliminar atributs de la base de dades. Aquesta operació també es pot realitzar amb el menú contextual del ratolí sobre la llista de variables de la base de dades.

- **Discretitzar Variable**: permet convertir una variable quantitativa creant-ne una de nova qualitativa a partir dels seus valors discretitzats. Permet discretitzar variables en intervals equidistants, de forma personalitzada o basada en boxplots. Aquesta operació també es pot realitzar amb el menú contextual del ratolí sobre la llista de variables de la base de dades.

En aquest menú també hi trobem les operacions per a tractar outliers i missings, estandaritzar variables o bé obtenir la distància mitja de la base de dades.

6.4. Executant algorismes

Un cop disposem de dades carregades i tractades, podrem començar a executar algorismes. Trobarem la funcionalitat de Gesconda II al menú superior, organitzada de manera que s'adapti al procés habitual d'extracció de coneixement, d'esquerra a dreta. Els menús de què disposa a Gesconda II són:

- **Rellevància d'Atributs**: Des d'aquí podrà executar diversos algorismes de Feature *Weighting* per tal de determinar de forma automàtica els pesos dels atributs de la base de dades.

- **Estadística descriptiva**: Permet realitzar anàlisi estadística univariant i bivariant amb variables qualitatives i quantitatives.

- **Gràfics**: permet la generació de diversos gràfics d'anàlisi.

- **Tècniques de Classificació**: en aquest menú hi trobarà tots els algorismes de clustering que ofereix Gesconda II, així com operacions per a validar-los

- **Inducció de Regles**: algorismes que generen regles de classificació. També hi ha opcions per a validar aquests algorismes, processar i exportar les regles generades.

- **Arbres de decisió**: permet generar arbres de decisió a partir de les dades que després podran ser visualitzats per al seu estudi a la vista corresponent.

- **Estadística predictiva**: En aquest menú s'hi troba la funcionalitat de Regressió lineal i l'Anàlisi de la variància d'un i dos factors.

7. Anàlisi econòmica del projecte

7.1. Recursos emprats

La realització d'aquest projecte no ha requerit de cap tipus de recurs material amb cost econòmic. Es requeria d'un lloc de treball equipat per al desenvolupador de l'aplicació i, a l'inici del projecte, es van plantejar dues opcions per a què no suposés cap cost. Per una banda, la facultat oferia aquest lloc a les seves instal·lacions, i per altra, que fou la opció escollida, el desenvolupador s'ofereix a utilitzar l'ordinador personal per raons de flexibilitat horària i comoditat, al estalviar desplaçaments.

7.2. Costos econòmics del projecte

El projecte va ser pressupostat amb 600 hores, que es reparteixen de la següent manera:

FASE	*Hores*	*Categoria*	*Preu*	*cost*
Anàlisi de requeriments	32	Analista	70 €/hora	2.240 €
Especificació	56	Analista	70 €/hora	3.920 €
Disseny	72	Analista	70 €/hora	5.040 €
Implementació	264	Programador	50 €/hora	13.200 €
Validació	72	Programador	50 €/hora	3.600 €
Documentació	144	Analista/Progr.	60 €/hora	8.640 €
TOTAL	640			36.640 €

Figura 7.1: Taula de recursos i costos econòmics del projecte

7.3. Anàlisi de desviacions

Un cop acabat el projecte podem fer un balanç del que realment ha costat el desenvolupament del projecte i d'aquesta manera, podem fer un anàlisi de com i perquè s'han produït desviacions.

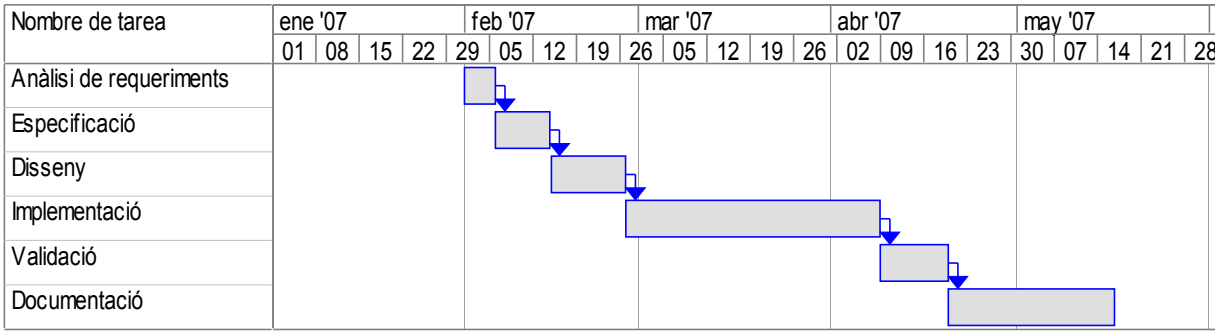


Figura 7.2: Planificació original del projecte

En la figura 7.2 es mostra el diagrama de gantt corresponent a les tasques del projecte determinades en el seu origen. Com es pot veure, la duració del projecte estava prevista en 3 mesos i mig. La realitat, però, no s'ajusta gaire a aquesta planificació degut a què la realització del projecte per part meva ha hagut de ser compaginada amb l'horari laboral habitual d'una altra feina.

Les tasques desenvolupades en aquest projecte han estat realitzades en horaris intempestius, caps de setmana i d'on ha estat possible treure una mica de temps per a la seva realització. Si no hagués estat per aquest motiu, el projecte hagués acabat amb temps suficient per a ser revisat i ampliat en algun aspecte. Al final, però s'han complert tots els requeriments inicials i s'ha pogut realitzar la presentació del projecte i d'aquesta memòria en el termini fixat com a data límit.

8. Conclusions

8.1. Concordança entre resultats i objectius

Amb el projecte finalitzat, és el moment de fer un petit *flashback* i fer un repàs dels requeriments inicials. Per cadascun d'ells comentem quina ha estat l'estratègia per assolir-lo i el resultat finalment obtingut.

8.1.1. Requeriments Tecnològics

Req.1.1. *Adaptació a Java5*

S'ha intentat aprofitar al màxim els avantatges que incorpora la versió 5 d'aquest llenguatge. A més d'eliminar l'ús de mètodes *deprecated*, s'ha refactoritzat el codi per tal d'utilitzar Generics en gairebé totes les col·leccions, de manera que s'evita l'ús innecessari de casts en el codi, i tot queda molt més net. A més es fan servir llibreries de parseig d'XML incorporades en l'estàndard a partir de la 1.4 que permeten la generació dinàmica dels menú o la persistència del model de dades en fitxer.

8.1.2. Requeriments d'integració

Req.1.1. *Integrar en la nova versió de Gesconda unificada, tots els mòduls existents*

Com s'ha pogut veure al llarg d'aquest document, el nou Gesconda II ofereix tota la funcionalitat dels mòduls que composaven el seu predecessor.

Req.1.2. *Unificació del model de dades*

Aquest requeriment era absolutament necessari i d'ell en depenien la resta. Sense un model unificat no hagués estat possible integrar la resta de funcions.

Req.1.3. *Unificació de la interfície d'usuari*

En l'apartat 4.4 es pot apreciar com ha quedat integrada la interfície d'usuari a Gesconda II

Req.1.4. *Eliminació de funcionalitat redundant*

Tan la discretització d'atributs com el tractament *d'outliers* i *missings* han estat unificats en un sol punt de l'aplicació.

Req.1.5. *Adaptació del codi de GESP*

Els algorismes de regressió, anova d'un i dos factors, així com les anàlisi estadístiques univariants i bivariants han estat integrades en el menú principal de Gesconda II

8.1.3. Requeriments de millora de la interfície gràfica d'usuari

Req.1.1. *Interfície gràfica ergonòmica*

En l'apartat 4.4 s'hi poden veure les funcionalitats que aporta la interfície gràfica d'usuari, així com les barres d'eines, dreceres i menús que incorpora Gesconda II. La disposició de els vistes en pestanyes també facilita la organització de l'espai de treball.

Req.1.2. *Visualització a la interfície de diferents paràmetres del sistema*

La interfície d'usuari mostra aquestes dades requerides a la barra d'estat a la part inferior.

Req.1.3. *Exportació de gràfics*

L'ús de la llibreria jFreeChart incorpora la funcionalitat de desar els gràfics en format png.

Req.1.4. *Distinció visual de valors outliers i missings a la matriu de dades*

La matriu de dades mostra en colors diferents aquests valors, i com a millora a la versió anterior no inclosa en els requeriments cal destacar que el càlcul d'aquests valors és completament dinàmic i no cal que l'usuari el demani per a veure els resultats. S'utilitza una tècnica de *lazy-loading* per a calcular la informació dels atributs.

Req.1.5. *Millorar la funcionalitat dels gràfics d'anàlisi de dades*

La llibreria jFreeChart, escollida per aquesta finalitat, acompleix de sobres amb aquest requeriment.

Req.1.6. *Representació gràfica dels arbres de decisió*

Per aquest requeriment ha calgut desenvolupar un component de visualització d'estructures arbòries personalitzat. La majoria de components d'aquest estil existents al mercat (tant opcions lliures com versions de pagament) s'especialitzen en mostrar els arbres en estructures de carpetes (similar a l'explorador de directoris) i no en jerarquies com les que genera Gesconda II.

8.1.4. Noves funcionalitats

Req.1.1. *Importació i exportació de dades en diferents formats*

Alguns dels mòduls existents ja realitzaven aquesta funcionalitat, però el codi estava repartit i amagat en punts difícils de trobar en l'aplicació. S'ha recuperat aquest codi, incorporat de nou per suportar algun format més, i ara està tot integrat al mateix menú.

Req.1.2. *Permetre l navegació i edició de valors sobre la matriu de dades*

La matriu de dades és totalment navegable i editable oferint barres de desplaçament quant és necessari. L'usuari pot reordenar les columnes de la taula en funció de les seves preferències.

Req.1.3. *Possibilitat de fer copy&paste dels valors de la matriu de dades amb altres aplicacions que gestionin matrius de dades*

S'han incorporat accions per a satisfer aquest requeriment i ara es poden realitzar aquestes operacions ja sigui des de la barra d'eines associada a cada taula o bé amb les típiques dreceres de teclat Ctrl+C i Ctrl+V

Req.1.4. *Permetre gestionar més de 2000 instàncies en una Base de Dades*

El model de dades fa ús exclusiu de memòria dinàmica, de manera que ara el límit de dades està marcat per la memòria física de la màquina on s'executa Gesconda.

Req.1.5. *Persistència dels resultats de l'aplicació*

S'ha dissenyat i desenvolupat un nou format de fitxer per a desar les dades de Gesconda II. Basat en tecnologia XML ara és possible guardar a més de la taula, els algorismes generats i altres valors, permetent que l'usuari pugui desar i restaurar una sessió de treball complerta.

Req.1.6. *Superposició de TS-Plot de varies variables*

Amb la nova llibreria gràfica es poden generar aquest nou tipus de gràfics.

Req.1.7. *Disposar d'un sistema d'ajuda a l'usuari*

L'aplicació proporciona una plataforma d'ajuda a l'usuari. La intenció d'aquesta plataforma és aportar a l'usuari de Gesconda, informació d'alt nivell sobre els algorismes que permet executar. Aquesta informació serà extreta directament de les memòries realitzades pels estudiants que van desenvolupar els algorismes. Per motius de temps, i perquè no formava part dels objectius del projecte, aquesta ajuda no conté la informació de tots els algorismes, però Gesconda II està preparat per a què sigui incorporada en quan es disposi d'ella.

8.1.5. Requeriments de facilitat de manteniment

Req.1.1. *Facilitar la modificació dels menús*

Els menús es generen dinàmicament a partir d'un fitxer XML, de fàcil manteniment, que permet personalitzar la ubicació de les accions, afegir-ne de noves o treure'n les que no es vulgui que apareguin.

Req.1.2. *Suport multi-idioma*

Els textos de l'aplicació estan desats en fitxers de propietats i no integrats en el codi. El programa accedeix al fitxer de propietats cada cop que necessita mostrar un text a l'usuari, de manera que replicant aquests fitxers i traduint-los s'aconsegueix el suport multi-idioma (l'aplicació fa ús de l'idioma que l'usuari té seleccionat al sistema operatiu). L'aplicació ve traduïda als idiomes català i castellà.

Req.1.3. *Generar documentació adequada per al posterior manteniment de l'aplicació*

Aquest document és precisament per aquest fi, però a més s'han incorporat comentaris en la major part del codi de manera que, amb l'ajuda de javadoc, s'ha generat la documentació complerta del codi i està a disposició en el CD de l'aplicació.

8.2. Treball futur

Per a la realització d'aquest projecte ha estat necessari consultar les memòries realitzades pels estudiants que van desenvolupar els mòduls de Gesconda I. En elles es mencionava que el treball futur era realitzar la integració dels mòduls que acabaven d'implementar. Ens trobem ara en un punt en què Gesconda II és una realitat i disposem d'una plataforma única per a la gestió de processos de mineria de dades.

No obstant, sempre cal anar endavant i cal plantejar des d'aquest mateix moment quines seran les millores per a la propera versió de Gesconda. Per una banda caldria començar el desenvolupament de la versió 3 d'aquest producte, però sense oblidar que la versió 2 acaba de sortir al mercat i que probablement requereixi de manteniment correctiu i evolutiu.

En un repositori de control de versions caldria obrir una branca per a la correcció de bugs i millores menors que generaria versions 2.1, 2.2 i successives, a mesura que es realitzin canvis. Mentre a la branca principal es podria començar el desenvolupament de la versió 3 que probablement inclouria molta més funcionalitat.

El feeling de l'usuari durant els primers mesos de posada en producció és molt important, de cara a realitzar correccions sobre la versió finalitzada però també per saber com plantejar i enfocar els nous desenvolupaments de la versió 3.

Com a proposta personal, la versió 3 podria incloure més algorismes, desenvolupats en la pròpia aplicació o bé incorporant-los d'altres sistemes externs com podria ser Weka o R. Gesconda podria oferir un tipus d'algorisme que fes de pont amb aquests sistemes i permetés l'execució de codi en ells, integrant els resultats en la pròpia aplicació.

La funcionalitat gràfica és ara molt més fàcil de millorar. La incorporació de la llibreria jFreeChart facilita la creació de gràfics a partir de les dades, de manera que dibuixar boxplots, gràfics en 3D, amb transparències o amb imatges incrustades mai ha estat

tan fàcil com ara. Caldria dissenyar noves vistes gràfiques i implementar-les utilitzant aquesta potent llibreria.

Una altra de les millores que podria aportar la propera versió de Gesconda és la possibilitat d'extreure les dades directament d'una Base de Dades Relacional, utilitzant *JDBC*, fet que permetria eliminar completament el problema de la memòria al treballar amb grans volums de dades. Sovint a més ens trobarem que les dades a tractar les generen aplicacions que treballen amb bases de dades relacionals, de manera que és una millora interessant obtenir les dades directament del lloc on resideixen.

La classe CMModel incorpora com hem vist abans una taula de dades amb les instàncies. Una altra millora a desenvolupar seria modificar aquesta classe per a què pugui treballar amb més d'una base de dades. En principi amb dues n'hi hauria d'haver prou, una de *training set* i una altra de real per aplicar els models dels algorismes i descobrir-hi informació. Amb això sí que aconseguiríem dur a terme tot el procés de mineria de dades des de l'aplicació Gesconda. En la versió actual no es pot implementar perquè caldria refer el codi d'alguns algorismes i quedava fora de l'abast del projecte.

8.3. Valoració personal

Com a valoració personal m'agradaria dir que realitzar aquest projecte en el moment en què vaig finalitzar les assignatures del pla d'estudis d'Enginyeria Informàtica (pla 94) no hagués estat possible. Si més no, els resultats no haurien estat els mateixos, perquè requeria que la persona que duia a terme el desenvolupament tingués uns coneixements avançats del llenguatge Java i de la tecnologia *swing*.

Finalitzar les assignatures del pla d'estudis, i entrar de ple al món professional sense realitzar el projecte fi de carrera, m'ha permès adquirir experiència en aquesta i altres branques de la professió. No hagués disposat d'aquesta experiència si hagués realitzat el projecte ara farà vuit anys, moment en què pràcticament vaig acabar les assignatures i vaig dedicar-me plenament al món professional.

Els coneixements de mineria de dades no han estat una dificultat per a mi, ja que, afortunadament, vaig tenir la sort de cursar una assignatura de lliure elecció titulada precisament Mineria de Dades, amb en Ricard Gavaldà. A més, les assignatures de la branca d'estadística i d'intel·ligència artificial també m'han aportat els coneixements necessaris per a la comprensió i realització d'aquest projecte. Ha calgut refrescar una mica el temari d'aquestes assignatures per a posar-me al nivell del requerit, però no ha estat una tasca excessivament complicada.

Crec sincerament que la preparació que he rebut en aquests anys de facultat ha estat excel·lent i de ben segur que si no fos per això, avui no estaria treballant en aquest sector ni realitzant tasques d'alt nivell tècnic.

Índex d'il·lustracions

Bibliografia

CLAVELL MARTINEZ, A. (2002). *Construcció d'un sistema inductiu de classificacions.* Memòria del Projecte. Facultat d'Informàtica de Barcelona. Universitat Politècnica de Catalunya

GARCIA BÒRDES, Mª.E. (2004). *Disseny i construcció d'un sistema inductiu de regles de classificacions.* Memòria del Projecte. Facultat d'Informàtica de Barcelona. Universitat Politècnica de Catalunya

MOLINA, L.C. (1998). *Data mining no processo d'extração de conhecimento de bases de daus*. Tesi de màster. São Carlos (Brasil): Institut de Ciências Matemáticas i Computação. Universitat de São Paulo.

Bibliografia consultada a la web:

http://alg.ncsa.uiuc.edu/tools/docs/d2k/manual/dataMining.html

http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.im.easy.doc/c_dm_process.html

http://www.crisp-dm.org/Process/index.htm

http://es.wikipedia.org/wiki/Scrum

http://java.sun.com/docs/books/tutorial/uiswing/index.html

Annex I: Llicència Gesconda – Termes i condicions

1. Aquesta llicència és aplicable al software Gesconda i a tota la documentació associada al projecte.

2. La llicència té un caràcter acadèmic i d'investigació i exclou l'autorització de qualsevol altre ús.

3. Gesconda podrà ser utilitzat lliurement per qualsevol persona que compleixi tots els termes i condicions inclosos en aquesta llicència. Podrà aconseguir el software prèvia sol·licitud als seus responsables, indicant les finalitats d'ús per garantir que es compleixen les condicions necessàries.

4. L'usuari de Gesconda es compromet a no fer ús lucratiu del producte, en tal cas caldrà posar-se en contacte amb els responsables de Gesconda per a negociar un altre tipus de llicència de caràcter comercial.

5. L'usuari de Gesconda es compromet a incloure una referència al software i al projecte en la publicació de qualsevol resultat obtingut mitjançant el propi producte, a més de notificar als responsables del producte de la publicació d'aquests articles.

6. Aquesta llicència exclou explícitament l'autorització del producte per a fins militars, d'espionatge, policials o que puguin suposar repressió a les persones.

7. Queda prohibida la cessió a tercers, així com la reproducció total o parcial del software Gesconda o de la seva documentació adjunta sense la autorització dels seus responsables.

8. Donat que el software Gesconda està llicenciat sense cap cost econòmic, es cedeix sense cap tipus de garantia ni responsabilitat per part dels responsables del projecte.

Annex II: Referència i llicències de llibreries usades

Gesconda utilitza diverses recursos lliures desenvolupats per altres autors. En aquest annex es citen i s'inclouen les llicències d'ús d'aquests recursos en compliment dels requeriments determinats per la pròpia llicència.

Llibreria Java per a la generació de gràfics a partir de les dades **jFreeChart**, sota llicència LGPL

Llibreria Java **jcommons** usada internament per jFreeChart, també sota llicència LGPL

Llibreria gràfica d'icones **Silk Icons** (http://www.famfamfam.com/lab/icons/silk/), sota llicència *Creative Commons Attribution 2.5*